



Red Hat OpenStack Platform 16.1

将 Overcloud 与现有 Red Hat Ceph 集群集成

配置 Overcloud 以使用 Stand-Alone Red Hat Ceph Storage

Red Hat OpenStack Platform 16.1 将 Overcloud 与现有 Red Hat Ceph 集群集成

配置 Overcloud 以使用 Stand-Alone Red Hat Ceph Storage

OpenStack Team
rhos-docs@redhat.com

法律通告

Copyright © 2023 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

本指南介绍了如何使用 Red Hat OpenStack Platform director 将 Overcloud 与现有的独立 Red Hat Ceph 集群集成。

目录

使开源包含更多	3
对红帽文档提供反馈	4
第 1 章 将 OVERCLOUD 与 CEPH STORAGE 集成	5
1.1. 关于 CEPH STORAGE	5
1.2. 使用外部 CEPHFS 部署共享文件系统服务	5
1.3. 配置 CEPH 对象存储以使用外部 CEPH 对象网关	6
第 2 章 准备 OVERCLOUD 节点	7
2.1. CEPH STORAGE 的部署前验证	7
2.2. 配置现有的 CEPH STORAGE 集群	7
2.3. 初始化 STACK 用户	9
2.4. 注册节点	9
2.5. 手动标记节点	11
第 3 章 与现有 CEPH STORAGE 集群集成	13
3.1. 安装 CEPH-ANSIBLE 软件包	13
3.2. 创建自定义环境文件	13
3.3. 将节点和类别分配给角色	15
3.4. RED HAT OPENSTACK PLATFORM 的 CEPH 容器及 CEPH STORAGE	15
3.5. 部署 OVERCLOUD	15
第 4 章 验证外部 CEPH STORAGE 集群集成	21
4.1. 收集 ID	21
4.2. 验证 CEPH STORAGE 集群	21
4.3. 对失败的验证进行故障排除	23
第 5 章 访问 OVERCLOUD	24

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。如需了解更多详细信息，请参阅 [CTO Chris Wright 信息](#)。 :leveloffset: +0

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。

使用直接文档反馈(DDF)功能

使用 **添加反馈** DDF 功能，用于特定句子、段落或代码块上的直接注释。

1. 以 *Multi-page HTML* 格式查看文档。
2. 请确定您看到文档右上角的 **反馈** 按钮。
3. 用鼠标指针高亮显示您想评论的文本部分。
4. 点 **添加反馈**。
5. 在**添加反馈**项中输入您的意见。
6. 可选：添加您的电子邮件地址，以便文档团队可以联系您以讨论您的问题。
7. 点 **Submit**。

第1章 将 OVERCLOUD 与 CEPH STORAGE 集成

Red Hat OpenStack Platform director 创建名为 overcloud 的云环境。您可以使用 director 为 overcloud 配置额外功能，如与 Red Hat Ceph Storage 集成。您可以将 overcloud 与创建的 Ceph Storage 集群与 director 或现有的 Ceph Storage 集群集成。

本指南包含有关如何将现有 Ceph Storage 集群与 overcloud 集成的说明。这意味着 director 将 overcloud 配置为使用 Ceph Storage 集群来满足存储需求。您可以在 overcloud 配置之外管理并扩展集群本身。

1.1. 关于 CEPH STORAGE

Red Hat Ceph Storage 是一个分布式数据对象存储，可提供性能、可靠性和可扩展性。分布式对象存储可容纳非结构化的数据，客户端可以同时使用现代对象接口和遗留接口。在每个 Ceph 部署的核心上，Ceph Storage 集群由多种类型的守护进程组成。以下是两种类型的守护进程：

Ceph OSD (Object Storage Daemon)

Ceph OSD 代表 Ceph 客户端存储数据。此外，Ceph OSD 使用 Ceph 节点的 CPU 和内存执行数据复制、重新平衡、恢复、监控和报告功能。

Ceph monitor

Ceph 监控器维护 Ceph 存储群集映射的主副本，且具有存储集群的当前状态。

有关 Red Hat Ceph Storage 的更多信息，请参阅 [Red Hat Ceph Storage 架构指南](#)。

1.2. 使用外部 CEPHFS 部署共享文件系统服务

Red Hat OpenStack Platform director 可以使用 CephFS 部署共享文件系统服务(manila)。CephFS 可以通过原生 CephFS 协议或通过 NFS 协议来消耗。

有关这些存储协议的更多信息，请参阅 [共享文件系统服务的 CephFS 后端指南](#) 中的 [Ceph 文件系统架构](#)。



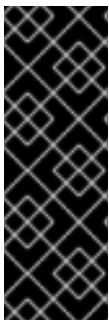
重要

您不能使用带有 CephFS 原生驱动程序的 Shared File Systems 服务(manila)通过 Manila CSI 向 Red Hat OpenShift Container Platform 提供共享。红帽不支持这种类型的部署。如需更多信息，请联系红帽支持团队。



重要

通过 NFS 使用 CephFS 的共享文件系统服务(manila)完全支持通过 Manila CSI 与 Red Hat OpenShift Container Platform 提供服务共享。这个解决方案不适用于大规模部署。有关重要建议，请参阅 <https://access.redhat.com/articles/6667651>。



重要

若要使用原生 CephFS 共享文件系统，客户端需要访问 Ceph 公共网络。当您将 overcloud 与现有的 Ceph 集群集成时，director 不会创建隔离存储网络来指定为 Ceph 公共网络。此网络假定为。不要直接访问 Ceph 公共网络，而是允许租户创建一个路由器来连接 Ceph 公共网络。

有关安全注意事项的更多信息，请参阅 [使用原生 CephFS 指南部署共享文件系统服务](#) 中的 [原生 CephFS 后端安全性](#)。

当您通过 NFS 协议使用 CephFS 时，director 会在由 Pacemaker (PCS)管理的 Controller 节点上部署 NFS-Ganesha 网关。PCS 通过一个主动 - 被动 (active-passive) 配置来管理集群可用性。



注意

此功能在 Ceph 4 的 Ceph Storage 4.1或更高版本，以及 Ceph 5 中的 Ceph Storage 5.0 或更高版本中被支持。您必须在 undercloud 上安装最新版本的 **ceph-ansible** 软件包。有关如何确定系统上安装的 Ceph Storage 版本的更多信息，请参阅 [Red Hat Ceph Storage 版本和相应的 Ceph 软件包版本](#)。

有关如何更新 undercloud 上的 **ceph-ansible** 软件包的更多信息，请参阅 [第 3.1 节 “安装 ceph-ansible 软件包”](#)。

前提条件

使用外部 Ceph Storage 集群配置共享文件系统服务需要满足以下先决条件：

- 外部 Ceph Storage 集群必须具有活跃的 MDS。
- 外部 Ceph Storage 集群必须具有 CephFS 文件系统，它基于 CephFS 数据的值 (**ManilaCephFSDataPoolName**)和 CephFS 元数据池 (**ManilaCephFSMetadataPoolName**)。更多信息请参阅 [第 3.2 节 “创建自定义环境文件”](#)。
- 外部 Ceph Storage 集群必须具有共享文件系统服务的 **cephx** 客户端名称和密钥。更多信息请参阅 [第 3.2 节 “创建自定义环境文件”](#)。

有关 Red Hat Ceph Storage 的更多信息，请参阅 [Red Hat Ceph Storage 文件系统指南](#)。

1.3. 配置 CEPH 对象存储以使用外部 CEPH 对象网关

Red Hat OpenStack Platform (RHOSP) Director 支持将外部 Ceph 对象网关(RGW)配置为对象存储服务。要使用外部 RGW 服务进行身份验证，您必须配置 RGW 以验证 Identity 服务(keystone)中的用户及其角色。

有关如何配置外部 Ceph 对象网关的更多信息，请参阅 *Using Keystone with the Ceph Object Gateway Guide* 中的 [Configuring the Ceph Object Gateway to use Keystone authentication](#)。

第 2 章 准备 OVERCLOUD 节点

在这种情况下部署的 overcloud 包括六个节点：

- 三个具有高可用性的 Controller 节点。
- 三个 Compute 节点。

director 将单独的 Ceph Storage 集群与自己的节点集成到 overcloud 中。您独立于 overcloud 管理此集群。例如，您可以使用 Ceph 管理工具扩展 Ceph Storage 集群，而不是通过 director 进行扩展。如需更多信息，请参阅 [Red Hat Ceph Storage](#) 文档库。

2.1. CEPH STORAGE 的部署前验证

为了帮助避免 overcloud 部署失败，请验证您的服务器上是否存在所需的软件包。

2.1.1. 验证 ceph-ansible 软件包版本

undercloud 包含基于 Ansible 的验证，您可以在部署 overcloud 前运行来识别潜在问题。这些验证可以帮助您避免 overcloud 部署失败，方法是在发生常见问题前识别它们。

流程

验证是否安装了 **ceph-ansible** 软件包的更正版本：

```
$ ansible-playbook -i /usr/bin/tripleo-ansible-inventory /usr/share/ansible/validation-playbooks/ceph-ansible-installed.yaml
```

2.1.2. 为预置备节点验证软件包

Ceph 只能服务具有一组特定软件包的 overcloud 节点。使用预置备节点时，您可以验证这些软件包是否存在。

有关预置备节点的更多信息，请参阅 [使用预置备节点配置基本的 overcloud](#)。

流程

验证服务器是否包含所需的软件包：

```
ansible-playbook -i /usr/bin/tripleo-ansible-inventory /usr/share/ansible/validation-playbooks/ceph-dependencies-installed.yaml
```

2.2. 配置现有的 CEPH STORAGE 集群

创建 OSD 池，定义功能，并为 Ceph Storage 集群创建密钥和 ID。

流程

1. 在与您的环境相关的 Ceph 集群中创建以下池：
 - **卷**：OpenStack Block Storage (cinder)的存储
 - **镜像**：OpenStack Image Storage (glance)的存储

- **虚拟机** : 实例的存储
- **备份** : 用于 OpenStack Block Storage Backup (cinder-backup)的存储
- **Metrics**: OpenStack Telemetry Metrics (gnocchi)存储
使用以下命令作为指南 :

```
[root@ceph ~]# ceph osd pool create volumes <_pgnum_>
[root@ceph ~]# ceph osd pool create images <_pgnum_>
[root@ceph ~]# ceph osd pool create vms <_pgnum_>
[root@ceph ~]# ceph osd pool create backups <_pgnum_>
[root@ceph ~]# ceph osd pool create metrics <_pgnum_>
```

如果您的 overcloud 部署由 CephFS 支持的共享文件系统服务(manila), 还要创建 CephFS 数据和元数据池 :

```
[root@ceph ~]# ceph osd pool create manila_data PGNUM
[root@ceph ~]# ceph osd pool create manila_metadata PGNUM
```

将 <_pgnum_> 替换为放置组的数量。每个 OSD 大约 100 个 PG 是最佳实践。例如, OSD 的总数乘以 100 倍, 除以副本数, **osd 池默认大小**。您还可以使用 [Ceph 放置组\(PG\)每个池计算器](#) 来确定合适的值。

2. 在 Ceph 集群中使用以下功能创建 **client.openstack** 用户 :

- **cap_mgr**: "allow *"
- **cap_mon**: profile rbd
- **cap_osd**: profile rbd pool=volumes, profile rbd pool=vms, profile rbd pool=images, profile rbd pool=backups, profile rbd pool=metrics
使用以下命令作为指南 :

```
[root@ceph ~]# ceph auth add client.openstack mgr 'allow *' mon 'profile rbd' osd 'profile
rbd pool=volumes, profile rbd pool=vms, profile rbd pool=images, profile rbd
pool=backups, profile rbd pool=metrics'
```

3. 注意为 **client.openstack** 用户创建的 Ceph 客户端密钥 :

```
[root@ceph ~]# ceph auth list
...
[client.openstack]
key = AQC+vYNXgDAgAhAAc8UoYt+OTz5uhV7ItLdwUw==
caps mgr = "allow *"
caps mon = "profile rbd"
caps osd = "profile rbd pool=volumes, profile rbd pool=vms, profile rbd pool=images, profile
rbd pool=backups, profile rbd pool=metrics"
...
```

示例中的 键值 **AQC+vYNXgDAgAhAAc8UoYt+OTz5uhV7ItLdwUw==** 是您 Ceph 客户端密钥。

4. 如果您的 overcloud 部署由 CephFS 支持的共享文件系统服务, 请在 Ceph 集群中使用以下功能创建 **client.manila** 用户 :

- `cap_mds` : 允许 *
- `cap_mgr` : 允许 *
- `cap_mon` : 允许 `r`, `allow command "auth del"`, `allow command "auth caps"`, `allow command "auth get"`, `allow command "auth get-or-create"`
- `cap_osd` : 允许 `rw` 使用下列命令作为指南 :

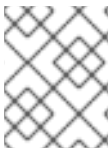
```
[root@ceph ~]# ceph auth add client.manila mon 'allow r, allow command "auth del",
allow command "auth caps", allow command "auth get", allow command "auth get-or-
create"' osd 'allow rw' mds 'allow *' mgr 'allow *'
```

5. 注意在 overcloud 部署模板中使用的 manila 客户端名称和键值 :

```
[root@ceph ~]# ceph auth get-key client.manila
AQDQ991cAAAAABAA0aXFrTnjH9aO39P0iVvYyg==
```

6. 记下 Ceph Storage 集群的文件系统 ID。在 `[global]` 部分中的 `fsid` 设置中使用集群配置文件中的 `fsid` 设置指定 :

```
[global]
fsid = 4b5c8c0a-ff60-454b-a1b4-9747aa737d19
...
```



注意

有关 Ceph Storage 集群配置文件的更多信息，请参阅 *Red Hat Ceph Storage 配置指南* 中的 [Ceph 配置](#)。

使用 Ceph 客户端密钥和文件系统 ID ; 以及以下流程中的 manila 客户端 IDS 和密钥 : [第 3.1 节 “安装 ceph-ansible 软件包”](#)

2.3. 初始化 STACK 用户

初始化 stack 用户，以配置用于访问 director CLI 工具的身份验证详情。

流程

1. 以 `stack` 用户身份登录 director 主机。
2. 输入以下命令来初始化您的 director 配置 :

```
$ source ~/stackrc
```

2.4. 注册节点

清单文件包含有关节点的硬件和电源管理详情。创建清单文件，以在 director 中配置和注册节点。

流程

1. 创建清单文件。使用节点定义模板 `instackenv.json` 示例作为参考 :

```
{
  "nodes":[
    {
      "mac":[
        "bb:bb:bb:bb:bb:bb"
      ],
      "cpu":"4",
      "memory":"6144",
      "disk":"40",
      "arch":"x86_64",
      "pm_type":"pxe_ipmitool",
      "pm_user":"admin",
      "pm_password":"p@55w0rd!",
      "pm_addr":"192.0.2.205"
    },
    {
      "mac":[
        "cc:cc:cc:cc:cc:cc"
      ],
      "cpu":"4",
      "memory":"6144",
      "disk":"40",
      "arch":"x86_64",
      "pm_type":"pxe_ipmitool",
      "pm_user":"admin",
      "pm_password":"p@55w0rd!",
      "pm_addr":"192.0.2.206"
    },
    {
      "mac":[
        "dd:dd:dd:dd:dd:dd"
      ],
      "cpu":"4",
      "memory":"6144",
      "disk":"40",
      "arch":"x86_64",
      "pm_type":"pxe_ipmitool",
      "pm_user":"admin",
      "pm_password":"p@55w0rd!",
      "pm_addr":"192.0.2.207"
    },
    {
      "mac":[
        "ee:ee:ee:ee:ee:ee"
      ],
      "cpu":"4",
      "memory":"6144",
      "disk":"40",
      "arch":"x86_64",
      "pm_type":"pxe_ipmitool",
      "pm_user":"admin",
      "pm_password":"p@55w0rd!",
      "pm_addr":"192.0.2.208"
    }
  ]
}
```

```

        "ff:ff:ff:ff:ff:ff"
    ],
    "cpu": "4",
    "memory": "6144",
    "disk": "40",
    "arch": "x86_64",
    "pm_type": "pxe_ipmitool",
    "pm_user": "admin",
    "pm_password": "p@55w0rd!",
    "pm_addr": "192.0.2.209"
  }
  {
    "mac": [
      "gg:gg:gg:gg:gg:gg"
    ],
    "cpu": "4",
    "memory": "6144",
    "disk": "40",
    "arch": "x86_64",
    "pm_type": "pxe_ipmitool",
    "pm_user": "admin",
    "pm_password": "p@55w0rd!",
    "pm_addr": "192.0.2.210"
  }
]
}

```

2. 将文件保存到 stack 用户的主目录：**/home/stack/instackenv.json**。
3. 初始化 stack 用户，然后将 **instackenv.json** 清单文件导入到 director：

```

$ source ~/stackrc
$ openstack overcloud node import ~/instackenv.json

```

openstack overcloud node import 命令可导入清单文件，并使用 director 注册每个节点。

4. 将内核和 ramdisk 镜像分配给每个节点：

```

$ openstack overcloud node configure <node>

```

结果

节点已在 director 中注册和配置。

2.5. 手动标记节点

注册每个节点后，您必须检查硬件并将节点标记为特定的配置集。使用 profile 标签将您的节点与类别匹配，然后将类别分配到部署角色。

流程

1. 触发硬件自省以检索每个节点的硬件属性：

```

$ openstack overcloud node introspect --all-manageable --provide

```

- **--all-manageable** 选项仅内省处于受管理状态的节点。在此示例中，所有节点都处于受管理状态。
- **--provide** 选项会在内省后将所有节点重置为 **主动** 状态。



重要

确保此过程成功完成。它可能需要 15 分钟来检查这些裸机节点。

2. 检索节点列表来识别它们的 UUID :

```
$ openstack baremetal node list
```

3. 在每个节点的 **properties/capabilities** 参数中添加 **profile** 选项，以便将节点手动标记到特定的配置集。添加 **profile** 选项会把节点标记为相关的配置集。
作为手动标记的替代，您可以配置 Automated Health Check (AHC)工具，以根据基准测试数据自动标记大量节点。

例如，要标记三个节点来使用 **control** 配置集，另一个三个节点使用 **compute** 配置集，创建以下配置集：

```
$ ironic node-update 1a4e30da-b6dc-499d-ba87-0bd8a3819bc0 add
properties/capabilities='profile:control,boot_option:local'
$ ironic node-update 6faba1a9-e2d8-4b7c-95a2-c7fbd12129a add
properties/capabilities='profile:control,boot_option:local'
$ ironic node-update 5e3b2f50-fcd9-4404-b0a2-59d79924b38e add
properties/capabilities='profile:control,boot_option:local'
$ ironic node-update 484587b2-b3b3-40d5-925b-a26a2fa3036f add
properties/capabilities='profile:compute,boot_option:local'
$ ironic node-update d010460b-38f2-4800-9cc4-d69f0d067efe add
properties/capabilities='profile:compute,boot_option:local'
$ ironic node-update d930e613-3e14-44b9-8240-4f3559801ea6 add
properties/capabilities='profile:compute,boot_option:local'
```


第 3 章 与现有 CEPH STORAGE 集群集成

要将 Red Hat OpenStack Platform 与现有 Ceph Storage 集群集成，您必须安装 **ceph-ansible** 软件包。之后，您可以创建自定义环境文件，并将节点和类别分配到角色。

3.1. 安装 CEPH-ANSIBLE 软件包

Red Hat OpenStack Platform director 使用 **ceph-ansible** 与现有 Ceph Storage 集群集成，但在 undercloud 上不默认安装 **ceph-ansible**。

流程

输入以下命令在 undercloud 上安装 **ceph-ansible** 软件包：

```
sudo dnf install -y ceph-ansible
```

3.2. 创建自定义环境文件

director 提供的参数到 **ceph-ansible**，以通过环境文件与外部 Ceph Storage 集群集成：

- **/usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-external.yaml**

如果您使用外部 CephFS 部署共享文件系统服务，则单独的环境文件会提供额外的参数。

- 对于原生的 CephFS，环境文件为 **/usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsnative-config.yaml**。
- 通过 NFS 使用 CephFS，环境文件是 **/usr/share/openstack-tripleo-heat-templates/environments/manila-cephfs-ganesha-config.yaml**。

director 在部署期间调用这些环境文件，以将现有的 Ceph Storage 集群与 overcloud 集成。有关详情请参阅 [第 3.5 节“部署 overcloud”](#)。

要配置集成，您必须将 Ceph Storage 集群的详细信息提供给 director。为此，请使用自定义环境文件来覆盖默认设置。

流程

1. 创建自定义环境文件：
/home/stack/templates/ceph-config.yaml

2. 在文件中添加 **parameter_defaults:** 部分：

```
parameter_defaults:
```

3. 使用 **parameter_defaults** 设置要在 **/usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-external.yaml** 中覆盖的所有参数。您必须至少设置以下参数：

- **CephClientKey**：Ceph 存储集群的 Ceph 客户端密钥。这是您在 [第 2.2 节“配置现有的 Ceph Storage 集群”](#) 中检索的键的值。例如，AQDLOh1VgEp6FRAAFzT7Zw+Y9V6JExQAsRnRQ==。

- **CephClusterFSID** : Ceph Storage 集群的文件系统 ID。这是您在 Ceph Storage 集群配置文件中检索到的 **fsid** 的值。第 2.2 节“配置现有的 Ceph Storage 集群”例如：**4b5c8c0a-ff60-454b-a1b4-9747aa737d19**。
- **CephExternalMonHost** : 以逗号分隔的 Ceph Storage 集群中所有 MON 主机的 IP 地址列表，如 **172.16.1.7, 172.16.1.8**。
例如：

```
parameter_defaults:
  CephClientKey: AQDLOh1VgEp6FRAAFzT7Zw+Y9V6JJExQAsRnRQ==
  CephClusterFSID: 4b5c8c0a-ff60-454b-a1b4-9747aa737d19
  CephExternalMonHost: 172.16.1.7, 172.16.1.8
```

4. 如有必要，请覆盖默认池名称或 Red Hat OpenStack Platform 客户端用户的名称以匹配您的 Ceph Storage 集群：

- **CephClientUserName: openstack**
- **NovaRbdPoolName: vms**
- **CinderRbdPoolName: volumes**
- **GlanceRbdPoolName: images**
- **CinderBackupRbdPoolName: backup**
- **GnocchiRbdPoolName: metrics**

5. 如果要部署由 CephFS 支持的共享文件系统服务，请设置数据和元数据池的名称：

```
ManilaCephFSDataPoolName: manila_data
ManilaCephFSMetadataPoolName: manila_metadata
```



注意

确保这些名称与您创建的池的名称匹配。

6. 设置您为 manila 创建的客户端密钥，以及该密钥的 Ceph 用户名称：

```
ManilaCephFSCephFSAuthId: 'manila'
CephManilaClientKey: 'AQDQ991cAAAAABAA0aXFrTnjH9aO39P0iVvYyg=='
```



注意

默认客户端用户名 **ManilaCephFSCephFSAuthId** 是 **manila**，除非您对其进行保护。始终需要 **CephManilaClientKey**。

7. 您还可以将 overcloud 参数添加到自定义环境文件。例如，要将 **vxlan** 设置为 **neutron** 网络类型，请将以下内容添加到 **parameter_defaults** 中：

```
NeutronNetworkType: vxlan
```

在创建自定义环境文件后，必须在部署 overcloud 时包括它。有关部署 overcloud 的更多信息，请参阅第 3.5 节“部署 overcloud”。

3.3. 将节点和类别分配给角色

规划 overcloud 部署涉及指定多少个节点以及分配给各个角色的类别。与所有 heat 模板参数一样，这些角色规格在自定义环境文件的 `parameter_defaults` 部分中声明，本例中为 `/home/stack/templates/ceph-config`。

要做到这一点，请使用以下参数：

表 3.1. overcloud 节点的角色和类别

Heat 模板参数	Description
ControllerCount	扩展的 Controller 节点数量
OvercloudControlFlavor	Controller 节点使用的 flavor (控制)
ComputeCount	扩展的 Compute 节点数量
OvercloudComputeFlavor	用于 Compute 节点的 flavor (计算)

例如，要将 overcloud 配置为为每个角色(Controller 和 Compute)部署三个节点，请将以下内容添加到 `parameter_defaults` 中：

```
parameter_defaults:
  ControllerCount: 3
  ComputeCount: 3
  OvercloudControlFlavor: control
  OvercloudComputeFlavor: compute
```



注意

有关更多信息和更完整的 heat 模板参数列表，请参阅 *Director 安装和使用 指南中的使用 CLI 工具创建 Overcloud*。

3.4. RED HAT OPENSTACK PLATFORM 的 CEPH 容器及 CEPH STORAGE

Ceph 容器需要配置 OpenStack Platform 来使用 Ceph，即使是外部 Ceph 集群。要与 Red Hat Enterprise Linux 8 兼容，Red Hat OpenStack Platform (RHOSP) 16 需要 Red Hat Ceph Storage 4。Ceph Storage 4 容器托管在 `registry.redhat.io`，这是需要身份验证的 registry。

您可以使用 heat environment 参数 `ContainerImageRegistryCredentials` 在 `registry.redhat.io` 中进行身份验证。如需更多信息，请参阅 [容器镜像准备参数](#)。

3.5. 部署 OVERCLOUD



注意

在 undercloud 安装过程中，在 `undercloud.conf` 文件中设置 `generate_service_certificate=false`。否则，必须在部署 overcloud 时注入信任定位器。有关如何注入信任定位器的更多信息，请参阅高级 Overcloud *自定义指南* 中的 [在 Overcloud 公共端点上启用 SSL/TLS](#)。

流程

- 创建 overcloud 需要 `openstack overcloud deploy` 命令的额外参数：

```
$ openstack overcloud deploy --templates \
  -e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-external.yaml \
  -e /home/stack/templates/ceph-config.yaml \
  -e --ntp-server pool.ntp.org \
```

这个示例命令使用以下选项：

- `--templates` - 从默认 heat 模板集合 `/usr/share/openstack-tripleo-heat-templates/` 创建 overcloud。
- `-e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-external.yaml` - 设置 director 以将现有的 Ceph 集群整合到 overcloud。
- `-e /home/stack/templates/ceph-config.yaml` - 添加自定义环境文件以覆盖 `-e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-external.yaml` 设定的默认值。在这种情况下，它是您在 [第 3.1 节 “安装 ceph-ansible 软件包”](#) 中创建的自定义环境文件。
- `--NTP-server pool.ntp.org` - 设置 NTP 服务器。

3.5.1. 为 CephFS 支持的共享文件系统服务添加额外的环境文件

如果部署使用 CephFS 支持的共享文件系统服务的 overcloud，您必须添加额外的环境文件。

流程

1. 使用以下选项之一创建并添加额外环境文件：

- 如果部署使用原生 CephFS 后端驱动程序的 overcloud，请使用 `/usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsnative-config.yaml`。
- 如果您通过 NFS 部署使用 CephFS 的 overcloud，请使用 `/usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsganesha-config.yaml`。
对于通过 NFS 的 CephFS，还必须部署自定义 Controller 角色，才能将 Ganesha CephFS 运行到 NFS 网关。此角色还配置隔离的 StorageNFS 网络，以向客户端提供共享。有关 StorageNFS 网络和自定义 Controller 角色的更多信息，请参阅 *CephFS via NFS Back End Guide for the Shared File Systems Service* 中的 [Deploying the updated environment](#)。

2. 根据您使用的 CephFS 后端，更改 `openstack overcloud deploy` 命令的形式。

- 对于原生 CephFS：

```
$ openstack overcloud deploy --templates \
  -e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-
```

```

ansible-external.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsnative-
config.yaml \
  -e /home/stack/templates/ceph-config.yaml \
  -e --ntp-server pool.ntp.org

```

- 通过 NFS 使用 CephFS :

```

$ openstack overcloud deploy --templates \
  -n /usr/share/openstack-tripleo-heat-templates/network_data_ganesha.yaml \
  -r /home/stack/custom_roles.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-
ansible-external.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsganesha-
config.yaml \
  -e /home/stack/templates/ceph-config.yaml \
  -e --ntp-server pool.ntp.org

```



注意

自定义 **ceph-config.yaml** 环境文件会覆盖 **ceph-ansible-external.yaml** 文件中的参数, 以及 **manila-cephfsnative-config.yaml** 文件或 **manila-cephfsganesha-config.yaml** 文件。因此, 在 **ceph-ansible-external.yaml** 和 **manila-cephfsnative-config.yaml** 或 **manila-cephfsganesha-config.yaml** 后的部署命令中包含自定义 **ceph-config.yaml** 环境文件。

环境文件示例

```

parameter_defaults:
  CinderEnableIscsiBackend: false
  CinderEnableRbdBackend: true
  CinderEnableNfsBackend: false
  NovaEnableRbdBackend: true
  GlanceBackend: rbd
  CinderRbdPoolName: "volumes"
  NovaRbdPoolName: "vms"
  GlanceRbdPoolName: "images"
  CinderBackupRbdPoolName: "backups"
  GnocchiRbdPoolName: "metrics"
  CephClusterFSID: <cluster_ID>
  CephExternalMonHost: <IP_address>,<IP_address>,<IP_address>
  CephClientKey: "<client_key>"
  CephClientUserName: "openstack"
  ManilaCephFSDataPoolName: manila_data
  ManilaCephFSMetadataPoolName: manila_metadata
  ManilaCephFSCephFSAuthId: 'manila'
  CephManilaClientKey: '<client_key>'
  ExtraConfig:
    ceph::profile::params::rbd_default_features: '1'

```

- 将变量 **< cluster_ID >**、**< IP_address >** 和 **< client_key >** 替换为适合您的环境的值。

3.5.2. 为外部 Ceph 对象网关(RGW)添加额外的环境文件以用于对象存储

如果部署使用已存在的 RGW 服务进行 Object storage 的 overcloud，您必须添加额外的环境文件。

流程

1. 将以下参数 **defaults** 添加到自定义环境文件，如 **swift-external-params.yaml**。替换要适合您的部署的值：

```
parameter_defaults:
  ExternalSwiftPublicUrl: 'http://<Public RGW endpoint or
loadbalancer>:8080/swift/v1/AUTH_%(project_id)s'
  ExternalSwiftInternalUrl: 'http://<Internal RGW endpoint>:8080/swift/v1/AUTH_%(project_id)s'
  ExternalSwiftAdminUrl: 'http://<Admin RGW endpoint>:8080/swift/v1/AUTH_%(project_id)s'
  ExternalSwiftUserTenant: 'service'
  SwiftPassword: 'choose_a_random_password'
```



注意

示例代码片段包含的参数值，可能与您在环境中使用的值不同：

- 远程 RGW 实例侦听的默认端口为 **8080**。端口可能会因配置外部 RGW 的配置而有所不同。
- overcloud 中创建的 **swift** 用户使用 **SwiftPassword** 参数所定义的密码。您必须将外部 RGW 实例配置为使用同密码相同的密码，以便使用 **rgw_keystone_admin_password** 进行身份服务进行身份验证。

2. 将以下代码添加到 Ceph 配置文件中，将 RGW 配置为使用 Identity 服务。替换变量值以适合您的环境：

```
rgw_keystone_api_version = 3
rgw_keystone_url = http://<public Keystone endpoint>:5000/
rgw_keystone_accepted_roles = member, Member, admin
rgw_keystone_accepted_admin_roles = ResellerAdmin, swiftoperator
rgw_keystone_admin_domain = default
rgw_keystone_admin_project = service
rgw_keystone_admin_user = swift
rgw_keystone_admin_password =
<password_as_defined_in_the_environment_parameters>
rgw_keystone_implicit_tenants = true
rgw_keystone_revocation_interval = 0
rgw_s3_auth_use_keystone = true
rgw_swift_versioning_enabled = true
rgw_swift_account_in_url = true
```



注意

director 默认在身份服务中创建以下角色和用户：

- rgw_keystone_accepted_admin_roles: ResellerAdmin, swiftoperator
- rgw_keystone_admin_domain: default
- rgw_keystone_admin_project: service
- rgw_keystone_admin_user: swift

3. 使用与部署相关的任何其他环境文件部署 overcloud：

```
openstack overcloud deploy --templates \
-e <your_environment_files>
-e /usr/share/openstack-tripleo-heat-templates/environments/swift-external.yaml
-e swift-external-params.yaml
```

3.5.3. 调用模板和环境文件

您还可以使用回答文件来调用所有模板和环境文件。例如，您可以使用以下命令部署相同的 overcloud：

```
$ openstack overcloud deploy \
--answers-file /home/stack/templates/answers.yaml \
--ntp-server pool.ntp.org
```

在这种情况下，该文件 **/home/stack/templates/answers.yaml** 包含：

```
templates: /usr/share/openstack-tripleo-heat-templates/
environments:
- /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible-external.yaml
\
- /home/stack/templates/ceph-config.yaml \
```

如需更多信息，请参阅 *Director 安装和使用指南* 中的 [将 overcloud 部署包含环境文件](#)。

3.5.4. OpenStack overcloud 部署命令选项

您可以输入以下命令查看可与 **openstack overcloud deploy** 命令一起使用的完整选项列表：

```
$ openstack help overcloud deploy
```

如需更多信息，请参阅 *Director 安装和使用指南* 中的 [使用 CLI 工具配置基本的 overcloud](#)。

3.5.5. 查看 overcloud 创建的状态

overcloud 创建过程开始，director 置备您的节点。完成此过程需要一些时间。

流程

要查看 overcloud 创建的状态，请以 **stack** 用户身份打开一个单独的终端并输入以下命令：

```
$ source ~/stackrc  
$ openstack stack list --nested
```


第 4 章 验证外部 CEPH STORAGE 集群集成

部署 overcloud 后，确认 Red Hat OpenStack Platform (RHOSP) 服务可以写入到 Ceph Storage 集群。



警告

RHOSP 不支持使用 Ceph 克隆格式 v2 或更高版本。从启用了 Ceph 克隆格式 v2 的 Ceph 集群中删除镜像或卷可能会导致意外行为和数据丢失。因此，请勿使用以下任一方法启用 Ceph 克隆格式 v2：

- 设置 **rbd 默认克隆格式 = 2**
- 运行 **ceph osd set-require-min-compat-client mimic**

4.1. 收集 ID

要验证您集成的 Ceph Storage 集群，您必须首先创建镜像、计算实例和一个卷并收集其对应的 ID。

流程

1. 使用镜像服务(glance)创建镜像。
有关如何创建镜像的更多信息，请参阅 [创建和管理镜像指南](#) 中的 [导入镜像](#)。
2. 记录 glance 镜像 ID，以备以后使用。
3. 创建计算(nova)实例。
有关如何创建实例的更多信息，请参阅 [创建和管理实例指南](#) 中的 [创建实例](#)。
4. 记录 nova 实例 ID，供以后使用。
5. 创建块存储(cinder)卷。
有关如何创建块存储卷的更多信息，请参阅 [存储指南](#) 中的 [创建卷](#)。
6. 记录 Cinder 卷 ID 以供以后使用。

4.2. 验证 CEPH STORAGE 集群

当您配置外部 Ceph Storage 集群时，您可以创建池和 **client.openstack** 用户来访问这些池。部署 overcloud 后，您可以使用包含 **client.openstack** 用户凭据的文件列出 Red Hat OpenStack Platform (RHOSP) 池的内容。

列出池的内容，并确认 Ceph Storage 集群上存在 glance 镜像、Compute 实例和 cinder 卷的 ID。

流程

1. 提供 undercloud 凭证：

```
[stack@undercloud-0 ~]$ source stackrc
```

- 列出可用的服务器以检索系统中节点的 IP 地址：

```
(undercloud) [stack@undercloud-0 ~]$ openstack server list

+-----+-----+-----+
| ID | Name | Status | Networks | Image | Flavor |
+-----+-----+-----+
| d5a621bd-d109-41ae-a381-a42414397802 | compute-0 | ACTIVE | ctlplane=192.168.24.31 | overcloud-full | compute |
| 496ab196-d6cb-447d-a118-5bafc5166cf2 | controller-0 | ACTIVE | ctlplane=192.168.24.37 | overcloud-full | controller |
| c01e730d-62f2-426a-a964-b31448f250b3 | controller-2 | ACTIVE | ctlplane=192.168.24.55 | overcloud-full | controller |
| 36df59b3-66f3-452e-9aec-b7e7f7c54b86 | controller-1 | ACTIVE | ctlplane=192.168.24.39 | overcloud-full | controller |
| f8f00497-246d-4e40-8a6a-b5a60fa66483 | compute-1 | ACTIVE | ctlplane=192.168.24.10 | overcloud-full | compute |
```

- 使用 SSH 登录到任何 Compute 节点：

```
(undercloud) [stack@undercloud-0 ~]$ ssh heat-admin@192.168.24.31
```

- 切换到 root 用户：

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

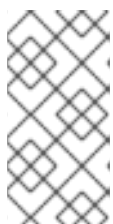
- 确认存在 `/etc/ceph/ceph.conf` 和 `/etc/ceph/ceph.client.openstack.keyring` 文件：

```
[root@compute-0 ~]# ls -l /etc/ceph/ceph.conf
-rw-r--r--. 1 root root 1170 Sep 29 23:25 /etc/ceph/ceph.conf
[root@compute-0 ~]# ls -l /etc/ceph/ceph.client.openstack.keyring
-rw-----. 1 ceph ceph 253 Sep 29 23:25 /etc/ceph/ceph.client.openstack.keyring
```

- 输入以下命令强制 `nova_compute` 容器使用 `rbd` 命令列出适当池的内容。

```
# podman exec nova_compute /usr/bin/rbd --conf /etc/ceph/ceph.conf --keyring
/etc/ceph/ceph.client.openstack.keyring --cluster ceph --id openstack ls vms
```

池名称必须与您配置 Ceph Storage 集群时创建的镜像、虚拟机和卷的池名称匹配。如需更多信息，请参阅[配置现有的 Ceph Storage 集群](#)。镜像、计算实例和卷的 ID 必须与您记录在 [第 4.1 节“收集 ID”](#) 中的 ID 匹配。



注意

`example` 命令带有 `podman exec nova_compute` 前缀，因为 `/usr/bin/rbd` 由 `ceph-common` 软件包提供，默认情况下不会在 `overcloud` 节点上安装。但是，它在 `nova_compute` 容器中可用。命令列出块设备镜像。有关更多信息，请参阅 [Ceph Storage 块设备指南](#) 中的 [列出块设备镜像](#)。

以下示例演示了如何使用 [第 4.1 节“收集 ID”](#) 中的 ID 确认每个池是否存在一个 ID。

```
# podman exec nova_compute /usr/bin/rbd --conf /etc/ceph/ceph.conf --keyring
/etc/ceph/ceph.client.openstack.keyring --cluster ceph --id openstack ls images | grep
4485d4c0-24c3-42ec-a158-4d3950fa020b

# podman exec nova_compute /usr/bin/rbd --conf /etc/ceph/ceph.conf --keyring
/etc/ceph/ceph.client.openstack.keyring --cluster ceph --id openstack ls vms | grep
64bcb731-e7a4-4dd5-a807-ee26c669482f

# podman exec nova_compute /usr/bin/rbd --conf /etc/ceph/ceph.conf --keyring
/etc/ceph/ceph.client.openstack.keyring --cluster ceph --id openstack ls volumes | grep
aeac15e8-b67f-454f-9486-46b3d75daff4
```

4.3. 对失败的验证进行故障排除

如果验证过程失败，请验证 **openstack.client** 用户和 Ceph Storage 监控 IP 或主机名的 Ceph 密钥可一起读取、写入和删除您为 Red Hat OpenStack Platform (RHOSP) 创建的 Ceph 存储池。

流程

1. 要缩短这个过程中必须执行的操作数量，请登录 Compute 节点，并为 **rbd** 命令创建一个别名：

```
# alias rbd="podman exec nova_compute /usr/bin/rbd --conf /etc/ceph/ceph.conf --keyring
/etc/ceph/ceph.client.openstack.keyring --cluster ceph --id openstack"
```

2. 确认您可以为新对象写入测试数据：

```
# rbd create --size 1024 vms/foo
```

3. 确认您可以看到测试数据：

```
# rbd ls vms | grep foo
```

4. 删除测试数据：

```
# rbd rm vms/foo
```



注意

如果这个过程失败，请联络您的 Ceph Storage 管理员以获得帮助。如果这个步骤成功，但无法创建计算实例、glance 镜像或 cinder 卷，请联系红帽支持团队。

第 5 章 访问 OVERCLOUD

director 会生成脚本来配置和验证 undercloud 与 overcloud 的交互。director 将此文件保存为 **stack** 用户主目录的 **overcloudrc** 文件。

流程

1. 输入以下命令使用 **overcloudrc** 文件：

```
$ source ~/overcloudrc
```

这会加载从 undercloud CLI 与 overcloud 交互所需的环境变量。

2. 要返回到 undercloud，请输入以下命令：

```
$ source ~/stackrc
```