Red Hat OpenStack Platform 17.1

配置持久性存储

配置和管理 OpenStack Block Storage、Object Storage 和 Shared File Systems 服务

Last Updated: 2024-05-23

Red Hat OpenStack Platform 17.1 配置持久性存储

配置和管理 OpenStack Block Storage、Object Storage 和 Shared File Systems 服务

OpenStack Team
rhos-docs@redhat.com

法律通告

摘要

本指南详细介绍了在 Red Hat OpenStack Platform 环境中使用和管理持久性存储的不同程序。它还包含配置和管理每种持久性存储类型的相应 OpenStack 服务的步骤。

目录

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 RED HAT OPENSTACK PLATFORM (RHOSP)中的持久性存储简介

在 Red Hat OpenStack Platform 中，存储由三个主要服务提供：

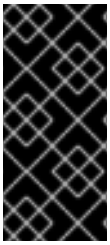
- Block Storage (**openstack-cinder**)
- Object Storage (**openstack-swift**)
- 共享文件系统存储(**openstack-manila**)

这些服务提供不同类型的持久性存储，它们在不同用例中都有自己的一组优点。本指南讨论了用于一般企业存储要求的适用性。

您可以使用 RHOSP 仪表板或命令行客户端管理云存储。您可以使用任一方法执行大多数步骤。但是，您只能在命令行中完成一些更高级的步骤。本指南为仪表板提供尽可能使用的步骤。

注意

有关 Red Hat OpenStack Platform 文档的完整套件，请参阅 Red Hat OpenStack Platform 文档。



重要

本指南记录了使用 **crudini** 应用一些自定义服务设置。因此，您需要首先安装 **crudini** 软件包：

```
# dnf install crudini -y
```

RHOSP 识别两种类型的存储： *ephemeral* 和 *persistent*。临时存储是仅与特定 Compute 实例关联的存储。实例终止后，即其临时存储。这种类型的存储对基本运行时要求很有用，如存储实例的操作系统。

持久性存储 的设计目的是独立于任何正在运行的实例的存活（持久性）。此存储用于需要被不同实例重复使用的任何数据，或超过特定实例的生命周期。RHOSP 使用以下类型的持久性存储：

卷

OpenStack Block Storage 服务(**openstack-cinder**)允许用户通过 *卷* 访问块存储设备。用户可以将卷附加到实例，以便使用通用持久性存储增强其临时存储。卷可以分离并重新附加到实例，只能通过它们所连接的实例访问。
您还可以将实例配置为使用临时存储。您可以配置块存储服务来将镜像写入卷，而不使用临时存储。然后，您可以使用卷作为实例的可引导根卷。

卷还通过备份和快照提供固有冗余和灾难恢复。另外，您还可以加密卷以提高安全性。

容器

OpenStack Object Storage 服务(openstack-swift)提供了一个完全分布式的存储解决方案，用于存储任何类型的静态数据或二进制对象，如介质文件、大型数据集和磁盘镜像。对象存储服务使用容器组织这些对象。
虽然卷的内容只能通过实例访问，但容器内的对象可以通过 Object Storage REST API 访问。因此，对象存储服务可以通过云中的几乎每个服务用作存储库。

共享

共享文件系统服务(**openstack-manila**)提供了轻松调配远程、可共享文件系统 *或共享 的方法*。共享允许云中的项目开放共享存储，并可同时被多个实例使用。

每种存储类型都设计为解决特定的存储要求。容器是为广泛访问而设计的，因此能够在所有存储类型中实现最高吞吐量、访问和容错能力。容器使用量更多地用于服务。

另一方面，卷主要用于实例消耗。它们不会享受与容器相同的访问和性能级别，但它们具有更大的功能集，并且具有比容器更原生的安全功能。共享与本例中的卷类似，但多个实例可以消耗它们。

以下小节讨论了在特定存储标准上下文中每个存储类型的架构和功能集。

1.1. 可扩展性和后端存储

通常，集群存储解决方案提供了更大的后端可扩展性。例如，当您将 Red Hat Ceph 用作 Block Storage (cinder)后端时，您可以通过添加更多 Ceph Object Storage Daemon (OSD)节点来扩展存储容量和冗余。Block Storage、Object Storage (swift)和共享文件系统服务(manila)服务支持 Red Hat Ceph Storage 作为后端。

块存储服务可以使用多种存储解决方案作为离散后端。在后端级别，您可以通过添加更多后端并重新启动服务来扩展容量。块存储服务还具有大量支持的后端解决方案列表，其中的一些功能具有额外的可扩展性功能。

默认情况下，对象存储服务使用配置的存储节点上的文件系统，并且可以使用尽可能多的空间。对象存储服务支持 XFS 和 ext4 文件系统，它们都可以扩展为消耗尽可能多的底层块存储。您还可以通过向存储节点添加更多存储设备来扩展容量。

共享文件系统服务从被一个或多个第三方后端存储系统管理的指定存储池中置备文件共享。您可以通过增加服务可用的大小或数量来扩展此共享存储，或者向部署添加更多第三方后端存储系统。

1.2. 存储可访问性和管理

卷只通过实例被消耗，一次只能附加到一个实例中并挂载到一个实例。用户可以创建卷快照，它们可用于克隆卷或将卷恢复到以前的状态。如需更多信息，请参阅 第 1.4 节 “存储冗余和灾难恢复”。作为项目管理员，您可以使用块存储服务 *创建卷类型*，它将聚合卷设置，如大小和后端。您可以将卷类型与 *服务质量 (QoS)* 规格相关联，以便为云用户提供不同的性能级别。您的用户可以指定创建新卷时所需的卷类型。例如，使用较高性能 QoS 规格的卷可能会为您的用户提供更多 IOPS，或者您的用户可以将更轻量的工作负载分配给使用较低性能 QoS 规格来节省资源的卷。

与卷一样，共享通过实例消耗。但是，共享可以直接挂载到实例中，不需要通过控制面板或 CLI 连接。共享也可以由多个实例同时挂载。共享文件系统服务还支持共享快照和克隆；您还可以创建 *共享类型*来聚合设置（与卷类型类似）。

容器中的对象可以通过 API 访问，并可供云内的实例和服务访问。这使得它们成为服务的对象存储库；例如，镜像服务(**openstack-glance**)将其镜像存储在由对象存储服务管理的容器中。

1.3. 存储安全性

块存储服务(cinder)通过卷加密提供基本数据安全性。因此，您可以将卷类型配置为通过静态密钥加密；然后，密钥用于加密从配置的卷类型创建的所有卷。更多信息请参阅 第 2.7 节 “Block Storage 服务 (cinder)卷加密”。

对象和容器安全性在服务和节点级别进行配置。Object Storage 服务(swift)没有为容器和对象提供原生加密。相反，对象存储服务优先选择云中可访问性的优先级，因此仅依赖于云网络安全来保护对象数据。

共享文件系统服务(manila)可以通过访问限制来保护共享，无论是实例 IP、用户或组还是 TLS 证书。此

外，一些共享文件系统服务部署可以功能独立的共享服务器来管理共享网络和共享之间的关系；一些共享服务器支持，甚至需要额外的网络安全性。例如，CIFS 共享服务器需要部署 LDAP、Active Directory 或 Kerberos 身份验证服务。

有关如何保护镜像服务(glance)的更多信息，如镜像签名和验证和元数据定义(metadef) API 限制，请参阅创建和管理 *镜像* 中的镜像服务(glance)。

1.4. 存储冗余和灾难恢复

块存储服务(cinder)具有卷备份和恢复，可为用户存储提供基本灾难恢复。使用备份来保护卷内容。该服务也支持快照。除了克隆外，您还可以使用快照将卷恢复到以前的状态。

在多后端环境中，您还可以在后端之间迁移卷。如果您需要使后端脱机进行维护，这非常有用。备份通常存储在独立于其源卷的存储后端中，以帮助保护数据。快照无法实现，因为快照取决于其源卷。

块存储服务还支持创建一致性组，将卷分组在一起，以便同时创建快照。这在多个卷之间提供了更高级别的数据一致性。更多信息请参阅 第 2.9 节 “块存储服务(cinder)一致性组”。

Object Storage 服务(swift)没有提供内置备份功能。您必须在文件系统或节点级别执行所有备份。对象存储服务具有更强大的冗余和容错能力，即使对象存储服务的最基本部署也多次复制对象。您可以使用 **dm-multipath** 等故障转移功能来增强冗余。

共享文件系统服务不提供共享的内置备份功能，但它允许您为克隆和恢复创建快照。

第 2 章 配置块存储服务(CINDER)

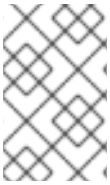
块存储服务(cinder)管理所有卷的管理、安全性、调度和整体管理。卷用作计算实例的主要持久性存储形式。

有关卷备份的更多信息，请参阅 *备份块存储卷* 指南。

重要

您必须在任何使用块存储服务和光纤通道(FC)后端的任何部署中安装主机总线适配器(HBA)。

块存储使用 Block Storage REST API 配置。



注意

确保您在使用 Block Storage REST API 版本 3，因为 Block Storage 不再支持版本 2。默认的 overcloud 部署通过设置环境变量 **OS_VOLUME_API_VERSION=3.0** 来为您完成此操作。

块存储 REST API 通过使用微版本来添加增强功能来保持向后兼容性。**cinder** CLI 使用 3.0 的 REST API 版本，除非您指定了特定的微版本。例如，要为 **cinder** 命令指定 3.17 微版本，请添加 **--os-volume-api-version 3.17** 参数。

注意

openstack CLI 只能使用 Block Storage REST API 版本 3.0，因为它不支持这些微版本。

2.1. 块存储服务后端

Red Hat OpenStack Platform (RHOSP)使用 director 部署。这样做有助于确保每个服务的正确配置，包括块存储服务(cinder)，以及扩展（扩展）后端。director 还有几个集成的后端配置。

默认情况下，块存储服务使用 LVM 后端作为卷的软件仓库。虽然此后端适合测试环境，但在生产环境中不支持 LVM。RHOSP 支持 Red Hat Ceph Storage 和 NFS 作为块存储服务后端。有关如何使用 RHOSP 部署 Red Hat Ceph Storage 的说明，请参阅 *部署 Red Hat Ceph Storage 和 OpenStack Platform* 。有关块存储及其配置支持的 NFS 的更多信息，请参阅配置 NFS 存储。

您还可以配置块存储服务以使用支持的第三方存储设备。director 包括部署不同后端解决方案所需的组件。

有关支持的块存储服务后端设备和驱动程序的完整列表，请参阅 *Red Hat OpenStack Platform 中的组件、插件和驱动程序支持* Cinder。所有第三方后端设备和驱动程序都有额外的部署指南。检查适当的部署指南，以确定后端设备或驱动程序是否需要插件。

如果将块存储配置为使用多个后端，您必须为每个后端创建一个卷类型。如果您在创建卷时没有指定后端，则块存储调度程序将使用过滤器来选择合适的后端。如需更多信息，请参阅配置默认块存储调度程序过滤器。

其他资源

- 创建和配置卷类型

2.1.1. 配置 NFS 存储

您可以将 overcloud 配置为使用共享 NFS 存储。

2.1.1.1. 支持的配置和限制

支持的 NFS 存储

- 红帽建议您使用经过认证的存储后端和驱动程序。红帽不推荐使用来自通用 NFS 后端的 NFS 存储，因为它的功能与经认证的存储后端和驱动程序相比是有限的。例如，通用 NFS 后端不支持卷加密和卷多重附加等功能。有关支持的驱动程序的详情，请查看 红帽生态系统目录。
- 对于 Block Storage (cinder)和计算(nova)服务，您必须使用 NFS 版本 4.0 或更高版本。Red Hat OpenStack Platform (RHOSP)不支持早期版本的 NFS。

不支持的 NFS 配置

- RHOSP 不支持 NetApp 功能 NAS 安全，因为它会干扰正常的卷操作。director 默认禁用该功能。因此，请不要编辑以下控制 NFS 后端还是 NetApp NFS 块存储后端是否支持 NAS 安全的 heat 参数：
  - **CinderNetappNasSecureFileOperations**
  - **CinderNetappNasSecureFilePermissions**
  - **CinderNasSecureFileOperations**
  - **CinderNasSecureFilePermissions**

使用 NFS 共享时的限制

- 当后端是 NFS 共享时，无法调整或重新构建具有交换磁盘的实例。

2.1.1.2. 配置 NFS 存储

您可以将 overcloud 配置为使用共享 NFS 存储。

流程

1. 创建一个环境文件来配置 NFS 存储，如 **nfs_storage.yaml**。
2. 在新环境文件中添加以下参数来配置 NFS 存储：

```
parameter_defaults:
  CinderEnableIscsiBackend: false
  CinderEnableNfsBackend: true
  GlanceBackend: file
  CinderNfsServers: 192.0.2.230:/cinder
  GlanceNfsEnabled: true
  GlanceNfsShare: 192.0.2.230:/glance
```

注意

不要配置 **CinderNfsMountOptions** 和 **GlanceNfsOptions** 参数，因为其默认值启用适合大多数 Red Hat OpenStack Platform (RHOSP)环境的 NFS 挂载选项。您可以在 **environments/storage/glance-nfs.yaml** 文件中看到 **GlanceNfsOptions** 参数的值。如果您在将多个服务配置为共享同一 NFS 服务器时遇到问题，请联系红帽支持。

3. 使用其他环境文件将 NFS 存储环境文件添加到堆栈中，并部署 overcloud：

```
(undercloud)$ openstack overcloud deploy --templates \
 -e [your environment files] \
 -e /home/stack/templates/nfs_storage.yaml
```

2.1.1.3. 配置外部 NFS 共享进行转换

当块存储服务(cinder)在 overcloud Controller 节点上执行镜像格式转换时，空间有限，大型镜像服务(glance)镜像的转换可能会导致节点根磁盘空间完全使用。您可以使用外部 NFS 共享进行转换，以防止完全填充节点上的空间。

有两个控制外部 NFS 共享配置的 director heat 参数：

- **CinderImageConversionNfsShare**
- **CinderImageConversionNfsOptions**

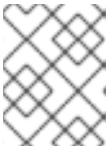
流程

1. 以 **stack** 用户身份登录 undercloud，再提供 **stackrc** 凭据文件。

```
$ source ~/stackrc
```

2. 在新的或现有与存储相关的环境文件中，添加关于外部 NFS 共享的信息。

```
parameter_defaults:
  CinderImageConversionNfsShare: 192.168.10.1:/convert
```



注意

用于控制 NFS 挂载选项的 **CinderImageConversionNfsOptions** 参数的默认值足以满足大多数环境。

3. 在 openstack overcloud deploy 命令中包含您的新配置的环境文件以及与您环境相关的任何其他环境文件。

```
$ openstack overcloud deploy \
--templates \
…
-e <existing_overcloud_environment_files> \
-e <new_environment_file> \
…
```

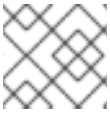
- 使用作为您现有部署一部分的环境文件列表替换 **<existing_overcloud_environment_files>**。
- 将 **<new_environment_file** > 替换为包含 NFS 共享配置的新或编辑的环境文件。

2.2. 块存储卷服务的高可用性

Block Storage 卷服务(**cinder-volume**)以主动-被动模式部署到 Controller 节点上。在这种情况下，Pacemaker 维护此服务的高可用性(HA)。

在分布式 Compute 节点(DCN)部署中，块存储服务以主动-被动模式部署到中央站点上。在这种情况下，Pacemaker 维护此服务的 HA。仅在需要存储的边缘站点上部署块存储卷服务。由于 Pacemaker 无法部署到边缘站点，因此块存储卷服务必须以主动-主动模式部署，以确保此服务的 HA。**dcn-storage.yaml** heat 模板执行此配置。在这种情况下，块存储服务被部署为块存储集群，通常在三个单独的主机上运行。

您需要管理块存储集群，这是所有配置的块存储卷服务分组，因此控制同一 Ceph 集群的相同池。如需更多信息，请参阅在边缘站点管理块存储集群。



注意

Red Hat Ceph Storage 后端的默认集群名称是 **tripleo@tripleo_ceph**。

2.2.1. 在边缘站点管理块存储集群

当您在需要存储的边缘站点以主动-主动模式部署块存储服务(**cinder-volume**)时，它将部署为块存储集群。此集群是所有配置相同的块存储卷服务的分组，因此控制同一 Ceph 集群的相同池。Red Hat Ceph Storage 后端的默认集群名称是 **tripleo@tripleo_ceph**。

您可以使用以下命令来管理此集群及其服务。

您可以维护集群。如需更多信息，请参阅 启动块存储集群维护。

注意

这些集群管理命令需要 3.17 或更高版本的 Block Storage (cinder) REST API 微版本。

用户目标	命令
要监控集群，请使用以下列：name、binary、state 和 status。 如果某些服务停机，请使用 **cinder --os-volume-api-version 3.17 service-list** 命令来确定受影响的服务。	**cinder --os-volume-api-version 3.17 cluster-list**
使用以下列（二进制、主机、区域、状态、状态、群集、禁用原因和集群名称）来确定关于集群的所有块存储服务的状态和详细信息。 **host** 列标识集群中运行的块存储卷服务。	**cinder --os-volume-api-version 3.17 service-list**

查看有关特定集群服务的详细信息。	**cinder --os-volume-api-version 3.17 cluster-show <clustered_service>** • 将 **<clustered_service>** 替换为集群服务的名称。
启用集群服务。	**cinder --os-volume-api-version 3.17 cluster-enable <clustered_service>**
禁用集群服务。	**cinderwagon-PROFILEos-volume-api-version 3.17 cluster-disable <clustered_service>**
列出可在块存储集群中管理的卷。如需更多信息，请参阅管理 和取消管理卷及其快照。	**cinder — os-volume-api-version 3.17 manageable-list --cluster <cluster_name>** • 将 **<cluster_name>** 替换为集群的名称。例如，tripleo@tripleo_ceph。
列出可在块存储集群中管理的快照。	**cinder — os-volume-api-version 3.17 snapshot-manageable-list --cluster <cluster_name>**
将非受管卷添加到块存储集群。	**cinderwagon-PROFILEos-volume-api-version 3.17 manage <unmanaged_volume> --cluster <cluster_name>** • 将 **<unmanaged_volume>** 替换为指定非受管卷所需的参数。

迁移块存储服务的卷。如需更多信息，请参阅使用 CLI 在后端间迁移卷。	**cinderwagon-PROFILEos-volume-api-version 3.17 migrate <volume> --cluster <cluster_name>** • 将 **<volume>** 替换为所需卷的名称或 ID。

2.2.2. 启动块存储集群维护

当在集群中分组了多个块存储卷服务(**cinder-volume**)时，必须至少运行其中一个服务来清理没有运行的服务。在这种情况下，您可以使用 **work-cleanup** 命令执行此集群维护。

注意

所有块存储卷服务在启动时执行自己的维护。

先决条件

- 您必须是一个项目管理员才能启动 Block Storage 集群维护。
- 块存储(cinder) REST API 微版本 3.24 或更高版本。

流程

1. 提供 overcloud 凭证文件：

```
$ source ~/<credentials_file>
```

   - 将 **<credentials_file>** 替换为您的凭据文件的名称，如 **overcloudrc**。

2. 运行以下命令，以验证块存储集群的所有服务是否正在运行：

```
$ cinder --os-volume-api-version 3.17 cluster-list --detailed
```

3. 如果有任何服务没有运行，请运行以下命令来识别这些特定服务：

```
$ cinder --os-volume-api-version 3.17 service-list
```

4. 运行以下命令来触发集群维护：

```
$ cinder --os-volume-api-version 3.24 work-cleanup --cluster <cluster_name>
```

   - 将 **<cluster_name>** 替换为集群的名称。例如，tripleo@tripleo_ceph。

2.3. 带有卷类型的组卷配置

使用 Red Hat OpenStack Platform，您可以创建卷类型，以便您可以将相关的设置应用到每种卷类型。您可以在创建卷前后分配所需的卷类型。如需更多信息，请参阅创建块存储卷 和 块存储卷调整。以下列表显示了您可以应用到卷类型的一些关联的设置：

- 卷的加密。如需更多信息，请参阅 块存储服务(cinder)卷加密。
- 卷使用的后端。如需更多信息，请参阅多个后端上的卷分配 和在后端间移动卷。
- 卷的相关服务质量(QoS)性能限制或 QoS 规格列表。如需更多信息，请参阅 块存储服务(cinder)服务质量规格。

设置使用名为 Extra Specs 的键值对与卷类型关联。当您在卷创建过程中指定卷类型时，块存储调度程序会将这些键值对作为设置应用。您可以将多个键值对与同一卷类型关联。

您可以创建卷类型，为您的云用户提供不同级别的性能：

- 为每个卷类型添加特定性能、弹性和其他 Extra Specs 作为键值对。
- 将不同的 QoS 性能限制或 QoS 规格与卷类型关联。

当您的用户创建卷时，他们可以选择适当的卷类型来满足其性能要求。

如果您创建卷且没有指定卷类型，则块存储将使用默认卷类型。您可以使用 Block Storage (cinder)配置文件来定义适用于所有项目（租户）的常规默认卷类型。但是，如果您的部署使用了特定于项目的卷类型，请确保为每个项目定义默认卷类型。在这种情况下，块存储使用特定于项目的卷类型，而不是常规的默认卷类型。如需更多信息，请参阅 定义特定于项目的默认卷类型。

其他资源

- 创建和配置卷类型
- 块存储服务后端
- 使用控制面板将 QoS 规格与卷类型关联
- 使用仪表板配置块存储服务卷加密

2.3.1. 列出后端驱动程序属性

与卷类型关联的属性使用名为 Extra Specs 的键值对。每个卷类型后端驱动程序都支持自己的一组 Extra Specs。有关支持驱动程序的 Extra Specs 的更多信息，请参阅后端驱动程序文档。

另外，您可以直接查询块存储主机，以列出其后端驱动程序的定义良好的标准 Extra Spec。

先决条件

- 您必须是一个项目管理员，才能直接查询块存储主机。

流程

1. 提供 overcloud 凭证文件：

```
$ source ~/<credentials_file>
```

   - 将 **<credentials_file** > 替换为您的凭据文件的名称，如 **overcloudrc**。

2. 确定 **cinder-volume** 的主机：

```
$ cinder service-list
```

此命令将返回一个列表，其中包含每个块存储服务的主机(**cinder-backup**、**cinder-scheduler** 和 **cinder-volume**)。例如：

```
+------------------+--------------------------+------+---------
|      Binary      |           Host           | Zone |  Status ...
+------------------+--------------------------+------+---------
|  cinder-backup   |   localhost.localdomain  | nova | enabled ...
| cinder-scheduler |   localhost.localdomain  | nova | enabled ...
|  cinder-volume   | *localhost.localdomain@lvm* | nova | enabled ...
+------------------+--------------------------+------+---------
```

3. 显示驱动程序功能，以确定块存储服务支持的 Extra Specs：

```
$ cinder get-capabilities <volsvchost>
```

- 将 **<volsvchost>** 替换为 **cinder-volume** 的主机。例如：

```
$ cinder get-capabilities localhost.localdomain@lvm
  +---------------------+----------------------------------------+
  |    Volume stats     |                Value                   |
  +---------------------+----------------------------------------+
  |     description     |                None                    |
  |     display_name    |                None                    |
  |    driver_version   |               3.0.0                    |
  |      namespace      | OS::Storage::Capabilities::localhost.loc...
  |      pool_name      |                None                    |
  |   storage_protocol  |               iSCSI                    |
  |     vendor_name     |            Open Source                 |
  |      visibility     |                None                    |
  | volume_backend_name |                lvm                     |
  +---------------------+----------------------------------------+
  +--------------------+----------------------------------------+
  | Backend properties |                Value                   |
  +--------------------+----------------------------------------+
  |    compression     |     {u'type': u'boolean', u'description'...
  |        qos         |      {u'type': u'boolean', u'des ...
  |    replication     |     {u'type': u'boolean', u'description'...
  | thin_provisioning  | {u'type': u'boolean', u'description': u'S...
  +--------------------+----------------------------------------+
```

Backend properties 列显示您可以设置的 Extra Spec Keys 列表，而 **Value** 列则提供有关有效对应值的信息。

2.3.2. 创建和配置卷类型

您可以创建卷类型，以便您可以将关联的设置应用到每种卷类型。

注意

如果将 Block Storage 服务(cinder)配置为使用多个后端，则必须为每个后端创建一个卷类型。

例如，您可以创建卷类型来为云用户提供不同的性能级别：

- 为每个卷类型添加特定性能、弹性和其他 Extra Specs 作为键值对。
- 将不同的 QoS 性能限制或 QoS 规格与卷类型关联。如需更多信息，请参阅 块存储服务(cinder) 服务质量规格。

当您的用户创建卷时，他们可以选择适当的卷类型来满足其性能要求。

先决条件

- 您必须是一个项目管理员，才能创建和配置卷类型。
- 访问 Red Hat OpenStack Platform (RHOSP) Dashboard (horizon)。如需更多信息，请参阅 *使用 OpenStack 控制面板管理云资源*。

流程

1. 以 admin 用户身份登录控制面板。
2. 选择 **Admin > Volumes > Volume Types**。
3. 单击**创建卷类型**。
4. 在 **Name** 字段中输入卷类型名称。
5. 单击**创建卷类型**。新类型会出现在 **卷类型** 表中。
6. 选择卷类型的 **View Extra Specs** 操作。
7. 点 **Create** 并指定 **Key** 和 **Value**。键值对必须有效；否则，在卷创建过程中指定卷类型将导致错误。例如，要为这个卷类型指定一个后端，请添加 **volume_backend_name** Key，并将 **Value** 设置为所需后端的名称。
8. 点 **Create**。关联的设置（键值对）现在出现在 **Extra Specs** 表中。

默认情况下，所有 OpenStack 项目都可以访问所有卷类型。如果您需要创建具有受限访问权限的卷类型，则需要通过 CLI 完成此操作。具体步骤请参阅创建和配置私有卷类型。

后续步骤

- 使用控制面板将 QoS 规格与卷类型关联

2.3.3. 编辑卷类型

编辑仪表板中的卷类型，以修改卷类型的 **Extra Specs** 配置。您还可以删除卷类型。

先决条件

- 您必须是项目管理员才能编辑或删除卷类型。
- 访问 Red Hat OpenStack Platform (RHOSP) Dashboard (horizon)。如需更多信息，请参阅 *使用 OpenStack 控制面板管理云资源*。

流程

1. 以 admin 用户身份登录控制面板。

2. 选择 **Admin > Volumes > Volume Types**。
3. 在 **Volume Types** 表中，选择卷类型的 **View Extra Specs** 操作。
4. 在此页的 **Extra Specs** 表中，您可以：
   - 在卷类型中添加一个新的设置。要做到这一点，点 **Create** 并指定您要与卷类型关联的新设置的键/值对。
   - 选择设置的 **Edit** action 来编辑与卷类型关联的现有设置。
   - 选择 extra specs 复选框并单击 **Delete Extra Specs** 和下一个对话框屏幕，删除与卷类型关联的现有设置。

要删除卷类型，请从**卷类型**表中选中其对应的复选框，然后点 **Delete Volume Types**。

2.3.4. 创建和配置私有卷类型

默认情况下，所有卷类型都可用于所有项目（租户）。您可以通过将其标记为 **私有** 来创建受限卷类型。为此，请将卷类型的 **is-public** 标志设为 **false**，因为此标志的默认值是 true。

私有卷类型可用于限制对具有特定属性的卷的访问。通常，这些设置应当仅可供特定项目使用。例如，正在测试的新后端或超高性能配置。

先决条件

- 您必须是一个项目管理员，才能创建、查看或配置私有卷类型的访问权限。

流程

1. 提供 overcloud 凭证文件：

```
$ source ~/<credentials_file>
```

   - 将 **<credentials_file>** 替换为您的凭据文件的名称，如 **overcloudrc**。

2. 创建新的 cinder 卷类型，并将 **is-public** 标志设置为 **false** ：

```
$ cinder type-create --is-public false <type_name>
```

   - 将 **<type_name>** 替换为您要调用这个新私有卷类型的名称。

默认情况下，私有卷类型只能被创建者访问。但是，管理员用户可使用以下命令查找并查看私有卷类型：

```
$ cinder type-list
```

此命令列出公共和私有卷类型的名称和 ID。您需要卷类型的 ID 来提供它的访问权限。

在项目级别授予私有卷类型的访问权限。因此，您需要知道所需项目的 ID。如果您不知道这个租户 ID，但您知道这个项目的名称，则运行：

注意

如果您不确定此用户名，**openstack user list** 命令将列出所有配置的用户的名称和 ID。

```
$ openstack user show <user_name>
```

- 将 &lt **;user_name** > 替换为所需项目用户的名称，以显示用户详情的列表，包括此用户关联的项目的 **tenantId**。

要授予项目对私有卷类型的访问权限，请运行：

```
$ cinder  type-access-add --volume-type <type_id> --project-id <tenant_id>
```

- 将 **<type_id>** 替换为所需的私有卷类型的 ID。
- 将 **<tenant_id** > 替换为所需的租户 ID。

要查看哪些项目可以访问私有卷类型，请运行：

```
$ cinder  type-access-list --volume-type <type_id>
```

要从私有卷类型的访问列表中删除项目，请运行：

```
$ cinder  type-access-remove --volume-type <type_id> --project-id <tenant_id>
```

2.3.5. 定义特定于项目的默认卷类型

可选：对于复杂部署，项目管理员可以为每个项目（租户）定义默认卷类型。

如果您创建卷且没有指定卷类型，则块存储将使用默认卷类型。

您可以使用 Block Storage (cinder) 配置文件 **cinder.conf** 的 **default_volume_type** 选项定义应用到所有项目的通用默认卷类型。

但是，如果您的 Red Hat OpenStack Platform (RHOSP)部署使用特定于项目的卷类型，请确保为每个项目定义默认卷类型。在这种情况下，块存储使用特定于项目的卷类型，而不是常规的默认卷类型。以下 RHOSP 部署示例需要特定于项目的默认卷类型：

- 跨越多个可用区(AZ)的分布式 RHOSP 部署。每个 AZ 都位于自己的项目中，并拥有自己的卷类型。
- 一个 RHOSP 部署，用于公司的三个不同部门。每个部门都位于自己的项目中，并拥有自己的专用卷类型。

先决条件

- 每个项目中至少有一个卷类型是特定于项目的默认卷类型。如需更多信息，请参阅 创建和配置卷类型。
- 块存储 REST API 微版本 3.62 或更高版本.
- 只有项目管理员才能为其项目定义、清除或列出默认卷类型。

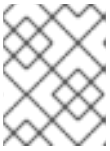
流程

1. 提供 overcloud 凭证文件：

```
$ source ~/<credentials_file>
```

- 将 **<credentials_file>** 替换为您的凭据文件的名称，如 **overcloudrc**。

2. 定义、清除或列出项目的默认卷类型：



注意

您必须将这些命令中的 **<project_id>** 替换为所需项目的 ID。若要查找每个租户的 ID 和名称，请运行 **openstack project list** 命令。

- 为项目定义默认卷类型：

```
$ cinder --os-volume-api-version 3.62 default-type-set <volume_type> <project_id>
```

  - 将 **<volume_type>** 替换为所需卷类型的名称或 ID。您可以运行 **cinder type-list** 命令来列出所有卷类型的名称和 ID。

- 清除项目的默认卷类型：

```
$ cinder --os-volume-api-version 3.62 default-type-unset <project_id>
```

- 列出项目的默认卷类型：

```
$ cinder --os-volume-api-version 3.62 default-type-list --project <project_id>
```

2.4. 为块存储服务(CINDER)创建和配置内部项目

有些块存储功能（如 Image-Volume 缓存）需要配置 *内部租户*。块存储服务使用此租户/项目来管理块存储项目，它们不一定需要公开给普通用户。此类项目的示例是缓存用于频繁卷克隆或迁移卷的临时副本的镜像。

流程

1. 要配置内部项目，首先创建一个名为 **cinder-internal** 的通用项目和用户。要做到这一点，登录到 Controller 节点并运行：

```
$ openstack project create --enable --description "Block Storage Internal Project" cinder-internal
  +-------------+----------------------------------+
  |  Property   |              Value               |
  +-------------+----------------------------------+
  | description |  Block Storage Internal Tenant   |
  |   enabled   |               True               |
  |      id     | cb91e1fe446a45628bb2b139d7dccaef |
  |     name    |         cinder-internal          |
  +-------------+----------------------------------+
$ openstack user create --project cinder-internal cinder-internal
  +----------+----------------------------------+
  | Property |              Value               |
  +----------+----------------------------------+
  |  email   |               None               |
  | enabled  |               True               |
  |    id    | 84e9672c64f041d6bfa7a930f558d946 |
  |   name   |         cinder-internal          |
```

```
|project_id| cb91e1fe446a45628bb2b139d7dccaef |
| username |        cinder-internal          |
+----------+---------------------------------+
```

2.5. 配置 IMAGE-VOLUME 缓存

块存储服务具有可选的 *Image-Volume 缓存*，可在从镜像创建卷时使用。此缓存旨在改进从频繁使用的镜像创建卷的速度。有关如何从镜像创建卷的详情，请参考 创建块存储卷。

启用后，Image-Volume 缓存会在从中创建卷时存储镜像的副本。此存储的镜像将缓存到块存储后端，以帮助在镜像下次用于创建卷时提高性能。Image-Volume 缓存的限制可以设置为大小（以 GB 为单位）、镜像数或两者。

几个后端支持 Image-Volume 缓存。如果您使用第三方后端，请参考其文档来获取有关 Image-Volume 缓存支持的信息。

先决条件

- 为块存储服务配置了 *内部租户*。如需更多信息，请参阅 为块存储服务(cinder)创建和配置内部项目。
- 已安装 undercloud。如需更多信息，请参阅使用 director *安装和管理 Red Hat OpenStack Platform 中的安装 director*。

流程

1. 以 **stack** 用户身份登录 undercloud 主机。
2. 查找 **stackrc** undercloud 凭证文件：

```
$ source ~/stackrc
```

3. 要在后端上启用和配置 Image-Volume 缓存，您必须在 overcloud 部署命令中包含的环境文件的 **ExtraConfig** 部分添加以下值：

```
parameter_defaults:
  ExtraConfig:
    cinder::config::cinder_config:
      DEFAULT/cinder_internal_tenant_project_id:
        value: TENANTID (1)
      DEFAULT/cinder_internal_tenant_user_id:
        value: USERID (2)
      BACKEND/image_volume_cache_enabled: (3)
        value: True
      BACKEND/image_volume_cache_max_size_gb:
        value: MAXSIZE (4)
      BACKEND/image_volume_cache_max_count:
        value: MAXNUMBER (5)
```

(1) 将 *TENANTID* 替换为 **cinder-internal** 项目的 ID。

(2) 将 *USERID* 替换为 **cinder-internal** 用户的 ID。

(3) 将 *BACKEND* 替换为目标后端的名称（特别是其 **volume_backend_name** 值）。

4 默认情况下，Image-Volume 缓存大小仅受后端限制。将 *MAXSIZE* 设置为所需的大小（以 GB 为单位）。

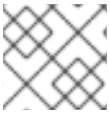
5 将 *MAXNUMBER* 设置为最大镜像数。

块存储服务数据库使用时间戳来跟踪每个缓存的镜像上次用于创建镜像。如果设置了 *MAXSIZE* 和 *MAXNUMBER*，则块存储服务将根据需要删除缓存的镜像，以便为新的镜像进行修改。每当满足 Image-Volume 缓存限制时，会首先删除带有旧时间戳的缓存镜像。

4. 将更新保存到环境文件中。
5. 使用其他环境文件将环境文件添加到堆栈中，并部署 overcloud。

2.6. 块存储服务(CINDER)服务质量规格

您可以将性能限制应用到云用户创建的卷，方法是将服务质量(QoS)规格与每种卷类型关联。例如，使用更高性能 QoS 规格的卷可为您的用户提供更多 IOPS，或者用户可将更轻量的工作负载分配给使用较低性能 QoS 规格来节省资源的卷。



注意

您必须是一个项目管理员，才能创建、配置、关联和解除关联 QoS 规格。

在创建 QoS 规格时，您必须选择所需的消费者。消费者决定您要应用 QoS 限制的位置，并确定哪些 QoS 属性键可用于定义 QoS 限制。有关可用消费者的更多信息，请参阅 QoS 规格的 Consumers。

您可以通过将所需的 QoS 属性键设置为特定于部署的值来创建卷性能限制。如需有关块存储服务(cinder)提供的 QoS 属性密钥的更多信息，请参阅 Block Storage QoS 属性键。

要创建 QoS 规格并将其与卷类型关联，请完成以下任务：

1. 创建并配置 QoS 规格。
2. 将 QoS 规格与卷类型关联。

您可以使用控制面板或使用 CLI 创建、配置和将 QoS 规格与卷类型关联。

2.6.1. QoS 规格的消费者

在创建 QoS 规格时，您必须选择所需的消费者。消费者决定您要应用 QoS 限制的位置，并确定哪些 QoS 属性键可用于定义 QoS 限制。Block Storage 服务(cinder)支持以下 QoS 规格的用户：

- **前端** ：计算服务(nova)在卷附加到实例时应用 QoS 限制。Compute 服务支持由块存储服务提供的所有 QoS 属性键。
- **后端** ：关联的卷类型的后端驱动程序应用 QoS 限制。每个后端驱动程序支持自己的一组 QoS 属性键。有关驱动程序支持的 QoS 属性密钥的更多信息，请参阅后端驱动程序文档。
  在不支持 **front-end** 消费者的情况下，应使用 **back-end** 消费者。例如，当通过裸机置备服务(ironic)将卷附加到裸机节点时。
- **两者** ：两个消费者尽可能应用 QoS 限制。因此，这个消费者类型支持以下 QoS 属性键：
  - 当卷附加到实例时，您可以使用 Compute 服务和后端驱动程序支持的每个 QoS 属性键。
  - 当卷没有附加到实例时，您只能使用后端驱动程序支持的 QoS 属性键。

2.6.2. 块存储 QoS 属性键

块存储服务为您提供了 QoS 属性键，以便您可以限制云用户创建的卷的性能。这些限制使用以下两个行业标准测量的存储卷性能：

- 每秒输入/输出操作(IOPS)
- 数据传输率，以每秒字节数为单位

QoS 规格的消费者决定支持哪些 QoS 属性键。如需更多信息，请参阅 QoS 规格的 Consumers。

块存储无法对 QoS 属性键执行错误检查，因为某些 QoS 属性键是由后端驱动程序外部定义的。因此，块存储会忽略任何无效的或不支持的 QoS 属性键。

重要

确保您正确拼写了 QoS 属性键。包含错误拼写属性键的卷性能限制将被忽略。

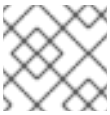
对于 IOPS 和数据传输速率测量，您可以配置以下性能限制：

修复了限制

通常，固定限制应定义卷性能测量的平均使用量。

burst 限制

通常，突发限制应该定义卷性能测量的意外活动期间。burst 限制允许增加特定时间的活动率，同时保持固定限制在平均使用量时较低。



注意

burst 限制所有使用突发长度 1 秒。

总限值

使用 **total** edServiceSet QoS 属性键为所需性能限制的读取和写入操作指定一个全局限制。

注意

除了使用总计限制，您可以将单独的限制应用到读写操作，或者选择仅限制读取或写入操作。

读取限制

使用 readjpeg QoS 属性键指定只适用于所需性能限制的 **读取操作** 的限制。

注意

当您指定总限制时，会忽略这个限制。

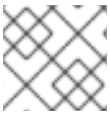
写入限制

使用 writejpeg QoS 属性键指定只适用于所需性能限制 **的写入操作** 的限制。

注意

当您指定总限制时，会忽略这个限制。

您可以使用以下 Block Storage QoS 属性键为部署创建卷性能限制：



注意

所有 QoS 属性键 **的默**认值为 0，这意味着限制是不受限制。

表 2.1. 块存储 QoS 属性键

性能限制	measurement 单元	QoS 属性键
修复了 IOPS	IOPS	**total_iops_sec** **read_iops_sec** **write_iops_sec**
修复了根据卷大小计算的 IOPS。 有关这些限制的使用限制的更多信息，请参阅 根据卷大小扩展的 QoS 限制。	IOPS 每 GB	**total_iops_sec_per_gb** **read_iops_sec_per_gb** **write_iops_sec_per_gb**
burst IOPS	IOPS	**total_iops_sec_max** **read_iops_sec_max** **write_iops_sec_max**
修复了数据传输速率	每秒字节数	**total_bytes_sec** **read_bytes_sec** **write_bytes_sec**
Burst 数据传输率	每秒字节数	**total_bytes_sec_max** **read_bytes_sec_max** **write_bytes_sec_max**
计算 IOPS 限制时 IO 请求的大小。 如需更多信息，请参阅为 IOPS 限制设置 IO 请求大小。	Bytes	**size_iops_sec**

2.6.2.1. 为 IOPS 限制设置 IO 请求大小

如果您实现 IOPS 卷性能限制，您还应指定典型的 IO 请求大小，以防止用户绕过这些限制。如果没有，用户可以提交多个大 IO 请求，而不是大量较小的请求。

使用 **size_iops_sec** QoS 属性键指定典型 IO 请求的最大大小（以字节为单位）。块存储服务使用此大小来计算提交的每个 IO 请求的比例 IO 请求，例如：

size_iops_sec=4096

- 8 KB 请求被计算为两个请求。
- 6 KB 请求计算为一和半个请求。
- 任何小于 4 KB 的请求都会被计算为一个请求。

块存储服务仅在计算 IOPS 限制时使用此 IO 请求大小限制。

注意

size_iops_sec 的默认值为 **0，** 它会在应用 IOPS 限制时忽略 IO 请求的大小。

2.6.2.2. IOPS 限制根据卷大小扩展

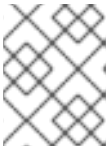
您可以创建 IOPS 卷性能限制，由用户创建的卷容量决定。这些服务质量(QoS)限制使用置备的卷的大小进行扩展。例如，如果卷类型的 IOPS 限制为 500 每个 GB 的卷大小，则此卷类型置备的 3 GB 卷会将读取 IOPS 限制为 1500。

重要

卷的大小在卷附加到实例时确定。因此，如果在将卷的大小附加到实例时更改了卷的大小，则只有在这个卷分离时，才会为新卷大小重新计算这些限制，然后重新附加到实例。

您可以使用以下 QoS 属性键（按 IOPS per GB 指定）来创建可扩展的卷性能限制：

- **total_iops_sec_per_gb** ：为读取和写入操作指定每个 GB 的全局 IOPS 限值。

  注意

  除了使用总计限制，您可以将单独的限制应用到读写操作，或者选择仅限制读取或写入操作。

- **read_iops_sec_per_gb** ：指定每个 GB 卷大小的 IOPS 限制，它只适用于读操作。

  注意

  当您指定总限制时，会忽略这个限制。

- **write_iops_sec_per_gb** ：指定每个 GB 卷大小的 IOPS 限制，它只适用于写入操作。

  注意

  当您指定总限制时，会忽略这个限制。

重要

包含这些 QoS 限制的 QoS 规格的消费者可以是 **前端**，也可以是 **两个**，但不能 **后端**。如需更多信息，请参阅 QoS 规格的 Consumers。

2.6.3. 使用仪表板创建和配置 QoS 规格

Quality of Service (QoS)规格是卷性能 QoS 限制列表。您可以通过将 QoS 属性键设置为部署特定值来创建每个 QoS 限制。要将 QoS 性能限制应用到卷，您必须将 QoS 规格与所需的卷类型关联。

先决条件

- 您必须是一个项目管理员，才能创建、配置、关联和解除关联 QoS 规格。
- 访问 Red Hat OpenStack Platform (RHOSP) Dashboard (horizon)。如需更多信息，请参阅 *使用 OpenStack 控制面板管理云资源*。

流程

1. 以 admin 用户身份登录控制面板。
2. 选择 **Admin > Volumes > Volume Types**。
3. 在 **QoS Specs** 表中，点 **Create QoS Spec**。
4. 输入 **QoS Spec** 的名称。
5. 在 **Consumer** 字段中，选择此 QoS 规格的消费者。如需更多信息，请参阅 QoS 规格的 Consumers。
6. 点 **Create**。新的 QoS 规格显示在 **QoS Specs** 表中。
7. 在 **QoS Specs** 表中，选择新 QoS 规格的 **Manage Specs** 操作，以打开 **Specs** 窗口，在其中添加 QoS 性能限制。
8. 点 **Specs** 窗口中的 **Create** 打开 **Create Extra Specs** 窗口。
9. 在 **Key** 字段中指定 QoS 性能限制的 QoS 属性键，并在 **Value** 字段中设置性能限制值。如需有关可用属性键的更多信息，请参阅 块存储 QoS 属性键。

重要

确保您正确拼写了 QoS 属性键。包含错误拼写属性键的卷性能限制将被忽略。

10. 点 **Create**，将 QoS 限制添加到您的 QoS 规格。
11. 对您要添加到 QoS 规格的每个 QoS 限制重复步骤 7 到 10。

后续步骤

- 使用控制面板将 QoS 规格与卷类型关联

2.6.4. 使用 CLI 创建和配置 QoS 规格

Quality of Service (QoS)规格是卷性能 QoS 限制列表。您可以通过将 QoS 属性键设置为部署特定值来创建每个 QoS 限制。要将 QoS 性能限制应用到卷，您必须将 QoS 规格与所需的卷类型关联。

先决条件

- 您必须是一个项目管理员，才能创建、配置、关联和解除关联 QoS 规格。

流程

1. 提供 overcloud 凭证文件：

```
$ source ~/<credentials_file>
```

   - 将 **<credentials_file** > 替换为您的凭据文件的名称，如 **overcloudrc**。

2. 创建 QoS 规格：

```
$ openstack volume qos create [--consumer <qos_spec_consumer>] <qos_spec_name>
```

   - 可选：将 **<qos_spec_consumer** > 替换为此 QoS 规格所需的消费者。如果未指定，则消费者默认为 **这两者**。如需更多信息，请参阅 QoS 规格的 Consumers。
   - 将 **<qos_spec_name** > 替换为 QoS 规格的名称。

3. 通过为每个要添加的 QoS 限制指定单独的 **--property <key=value&gt**; 参数，为 QoS 规格添加性能限制：

```
$ openstack volume qos set --property <key>=<value> <qos_spec_name>
```

   - 将 **<key** > 替换为所需性能约束的 QoS 属性键。如需更多信息，请参阅 块存储 QoS 属性键。

重要

确保您正确拼写了 QoS 属性键。包含错误拼写属性键的卷性能限制将被忽略。

   - 在 QoS 属性键所需的测量单元中，将 < **value** > 替换为此性能约束的部署特定限制。
   - 将 **<qos_spec_name** > 替换为 QoS 规格的名称或 ID。
     Example:

```
$ openstack volume qos set \
 --property read_iops_sec=5000 \
 --property write_iops_sec=7000 \
 myqoslimits
```

4. 查看 QoS 规格：

```
$ openstack volume qos list
+--------------------------------------+---------+-----------+--------------+-------------------------------
--------------------+
| ID                                   | Name    | Consumer  | Associations | Properties
|
+--------------------------------------+---------+-----------+--------------+-------------------------------
```

```
--------------------+
| 204c6ba2-c67c-4ac8-918a-03f101811235 | myqoslimits | front-end |              |
read_iops_sec='5000', write_iops_sec='7000' |
+--------------------------------------+---------+-----------+--------------+--------------------------------
--------------------+
```

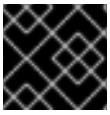
此命令提供了所有配置的 QoS 规格的配置详情表。

后续步骤

- 使用 CLI 将 QoS 规格与卷类型关联

2.6.5. 使用控制面板将 QoS 规格与卷类型关联

您必须将服务质量(QoS)规格与现有卷类型关联，才能将 QoS 限制应用到卷。



重要

如果卷已附加到实例，则 QoS 限制仅在卷分离时应用于此卷，然后重新附加到这个实例。

先决条件

- 您必须是一个项目管理员，才能创建、配置、关联和解除关联 QoS 规格。
- 创建所需的卷类型。如需更多信息，请参阅 创建和配置卷类型。
- 创建了所需的 QoS 规格。如需更多信息，请参阅使用仪表板创建和配置 QoS 规格。
- 访问 Red Hat OpenStack Platform (RHOSP) Dashboard (horizon)。如需更多信息，请参阅 *使用 OpenStack 控制面板管理云资源*。

流程

1. 以 admin 用户身份登录控制面板。
2. 选择 **Admin > Volumes > Volume Types**。
3. 在 **Volume Types** 表中，选择所需卷类型的 **Manage QoS Spec Association** 操作。
4. 从 **要关联的 QoS Spec 列表中选择所需的 QoS**规格。
5. 单击 **关联**。QoS 规格添加到编辑 **卷类型的关联 QoS Spec** 列中。

2.6.6. 使用 CLI 将 QoS 规格与卷类型关联

您必须将服务质量(QoS)规格与现有卷类型关联，才能将 QoS 限制应用到卷。

重要

如果卷已附加到实例，则 QoS 限制仅在卷分离时应用于此卷，然后重新附加到这个实例。

先决条件

- 您必须是一个项目管理员，才能创建、配置、关联和解除关联 QoS 规格。

- 创建所需的卷类型。如需更多信息，请参阅 创建和配置卷类型。
- 创建了所需的 QoS 规格。如需更多信息，请参阅使用 CLI 创建和配置 QoS 规格。

流程

1. 提供 overcloud 凭证文件：

```
$ source ~/<credentials_file>
```

   - 将 **<credentials_file** > 替换为您的凭据文件的名称，如 **overcloudrc**。

2. 将所需的 QoS 规格与所需的卷类型关联：

```
$ openstack volume qos associate <qos_spec_name> <volume_type>
```

   - 将 **<qos_spec_name** > 替换为 QoS 规格的名称或 ID。您可以运行 **openstack volume qos list** 命令来列出所有 QoS 规范的名称和 ID。
   - 将 **<volume_type** > 替换为卷类型的名称或 ID。您可以运行 **cinder type-list** 命令来列出所有卷类型的名称和 ID。

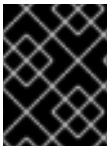
3. 验证 QoS 规格是否已关联：

```
$ openstack volume qos list
```

   输出表的 **Associations** 列显示哪些卷类型与这个 QoS 规格相关联。

2.6.7. 使用控制面板从卷类型取消关联 QoS 规格

当您不再需要将 QoS 限制应用到该卷类型的卷时，您可以将服务质量(QoS)规格与卷类型解除关联。



重要

如果卷已附加到实例，则 QoS 限制仅在卷分离时从这个卷中删除，然后重新附加到这个实例。

先决条件

- 您必须是一个项目管理员，才能创建、配置、关联和解除关联 QoS 规格。
- 访问 Red Hat OpenStack Platform (RHOSP) Dashboard (horizon)。如需更多信息，请参阅 *使用 OpenStack 控制面板管理云资源*。

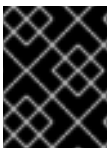
流程

1. 以 admin 用户身份登录控制面板。
2. 选择 **Admin > Volumes > Volume Types**。
3. 在 **Volume Types** 表中，选择所需卷类型的 **Manage QoS Spec Association** 操作。
4. 从 **QoS Spec to be associated** 列表中选择 **None**。
5. 单击 **关联**。

QoS 规范应从编辑 **的卷类型的关联 QoS Spec** 列中删除。

2.6.8. 使用 CLI 将 QoS 规格与卷类型关联

当您不再需要将 QoS 限制应用到该卷类型的卷时，您可以将服务质量(QoS)规格与卷类型解除关联。



重要

如果卷已附加到实例，则 QoS 限制仅在卷分离时从这个卷中删除，然后重新附加到这个实例。

先决条件

- 您必须是一个项目管理员，才能创建、配置、关联和解除关联 QoS 规格。

流程

1. 提供 overcloud 凭证文件：

```
$ source ~/<credentials_file>
```

   - 将 **<credentials_file** > 替换为您的凭据文件的名称，如 **overcloudrc**。

2. 解除与 QoS 规格关联的卷类型。当多个卷类型与相同的 QoS 规格关联时，您可以解除关联特定卷类型，或者所有卷类型：

   - 解除与 QoS 规格关联的特定卷类型：

```
$ openstack volume qos disassociate <qos_spec_name> --volume-type <volume_type>
```

     - 将 **<qos_spec_name** > 替换为 QoS 规格的名称或 ID。您可以运行 **openstack volume qos list** 命令来列出所有 QoS 规范的名称和 ID。
     - 将 **<volume_type** > 替换为与此 QoS 规格关联的卷类型名称或 ID。您可以运行 **cinder type-list** 命令来列出所有卷类型的名称和 ID。

   - 解除与 QoS 规格关联的所有卷类型：

```
$ openstack volume qos disassociate <qos_spec_name> --all
```

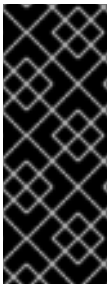
3. 验证 QoS 规格是否已解除关联：

```
$ openstack volume qos list
```

   此 QoS 规范的 **Associations** 列不应指定卷类型或为空。

2.7. BLOCK STORAGE 服务(CINDER)卷加密

卷加密有助于提供基本数据保护，以防卷后端被破坏或 outright stolen。Compute 和 Block Storage 服务都集成在一起，以允许实例读取访问和使用加密卷。您必须部署 Barbican 才能利用卷加密。



重要

- 基于文件的卷（如 NFS）不支持卷加密。
- 卷加密只支持 LUKS1 而不是 LUKS2。
- 不支持将未加密的卷重新设置为相同大小的加密卷，因为加密卷需要额外的空间来存储加密数据。有关加密未加密的卷的更多信息，请参阅 加密未加密的卷。

卷加密通过卷类型应用。有关加密卷类型的详情，请参考 使用控制面板配置块存储服务卷加密 或使用 CLI 配置块存储服务卷加密。

如需更多信息，请参阅使用 OpenStack Key Manager (barbican)管理您的块存储(cinder)加密密钥，请参阅 加密块存储(cinder)卷。

2.7.1. 使用仪表板配置块存储服务卷加密

要创建加密的卷，您首先需要一个 *加密的卷类型*。加密卷类型涉及设置它应使用的供应商类、密码和密钥大小。您还可以重新配置加密卷类型的加密设置。

您可以调用加密的卷类型来自动创建加密的卷。

先决条件

- 您必须是项目管理员才能创建加密卷。
- 访问 Red Hat OpenStack Platform (RHOSP) Dashboard (horizon)。如需更多信息，请参阅 *使用 OpenStack 控制面板管理云资源*。

流程

1. 以 admin 用户身份登录控制面板。
2. 选择 **Admin > Volumes > Volume Types**。
3. 在要加密的卷的 **Actions** 列中，选择 **Create Encryption** 以启动 **Create Volume Type Encryption** 向导。
4. 在这里，配置卷类型加密的 **Provider**,**Control Location**,**Cipher**, 和 **Key Size** 设置。**Description** 列描述了每个设置。

重要

以下列出的值是 **Provider**、**Cipher** 和 **Key Size** 唯一支持的选项。

   a. 为 **Provider** 输入 **luks**。
   b. 为 **Cipher** 输入 **aes-xts-plain64**。
   c. 为 **Key Size** 输入 **256**。

5. 单击 **Create Volume Type Encryption**。

您还可以重新配置加密卷类型的加密设置。

1. 从卷类型的 **Actions** 列中选择 **Update Encryption** 以启动 **Update Volume Type Encryption**向导。
2. 在 **Project > Compute > Volumes**中，检查 **Volumes** 表中的 **Encrypted** 列，以确定卷是否加密。
3. 如果卷加密，请单击该列中的 **Yes** 以查看加密设置。

其他资源

- 使用 CLI 配置块存储服务卷加密

2.7.2. 使用 CLI 配置块存储服务卷加密

要创建加密的卷，您首先需要一个 *加密的卷类型*。加密卷类型涉及设置它应使用的供应商类、密码和密钥大小。

先决条件

- 您必须是项目管理员才能创建加密卷。

流程

1. 提供 overcloud 凭证文件：

```
$ source ~/<credentials_file>
```

   - 将 **<credentials_file >** 替换为您的凭据文件的名称，如 **overcloudrc**。

2. 创建卷类型：

```
$ cinder type-create myEncType
```

3. 配置密码、密钥大小、控制位置和提供程序设置：

```
$ cinder encryption-type-create --cipher aes-xts-plain64 --key-size 256 --control-location
front-end myEncType luks
```

4. 创建一个加密卷：

```
$ cinder --debug create 1 --volume-type myEncType --name myEncVol
```

2.7.3. 自动删除卷镜像加密密钥

将加密卷上传到镜像服务(glance)时，块存储服务(cinder)在密钥管理服务(barbican)中创建一个加密密钥。这会在加密密钥和存储的镜像之间创建一个 1:1 关系。

加密密钥删除可防止密钥管理服务无限度地消耗资源。块存储、密钥管理和镜像服务自动管理加密卷的密钥，包括删除密钥。

块存储服务会自动将两个属性添加到卷镜像中：

- **cinder_encryption_key_id** - 密钥管理服务为特定镜像存储的加密密钥的标识符。

- **cinder_encryption_key_deletion_policy** - 告知镜像服务是否删除与此镜像关联的密钥的策略。

重要

这些属性的值会被自动分配。为**避免意外的数据丢失，请不要调整这些值。**

当您创建卷镜像时，块存储服务会将 **cinder_encryption_key_deletion_policy** 属性设置为 **on_image_deletion**。当您删除卷镜像时，如果 **cinder_encryption_key_deletion_policy** 等于 **on_image_deletion** _policy，则镜像服务会删除对应的加密密钥。

重要

红帽不推荐手动操作 **cinder_encryption_key_id** 或 **cinder_encryption_key_deletion_policy** 属性。如果您使用由 **cinder_encryption_key_id** 值标识的加密密钥用于任何其他目的，则风险数据丢失。

2.8. 为块存储卷后端部署可用区

可用域是特定于供应商的，对云实例和服务进行分组的方法。director 使用 **CinderXXXAvailabilityZone** 参数（其中 **XXX** 与特定后端相关联）为块存储卷后端配置不同的可用区。

先决条件

- 已安装 undercloud。如需更多信息，请参阅使用 director *安装和管理 Red Hat OpenStack Platform 中的安装 director*。

流程

1. 以 **stack** 用户身份登录 undercloud 主机。
2. 查找 **stackrc** undercloud 凭证文件：

```
$ source ~/stackrc
```

3. 在环境文件中添加以下参数来创建两个可用区：

```
parameter_defaults:
 CinderXXXAvailabilityZone: zone1
 CinderYYYAvailabilityZone: zone2
```

- 将 **XXX** 和 **YYY** 替换为支持的后端值，例如：

```
CinderISCSIAvailabilityZone
CinderNfsAvailabilityZone
CinderRbdAvailabilityZone
```

注意

在 **/usr/share/openstack-tripleo-heat-templates/deployment/cinder/** 目录中搜索与后端关联的 heat 模板，以获取正确的后端值。

以下示例部署两个后端，其中 **rbd** 是 zone 1， **iSCSI** 是 zone 2：

```
parameter_defaults:
 CinderRbdAvailabilityZone: zone1
 CinderISCSIAvailabilityZone: zone2
```

4. 将更新保存到环境文件中。
5. 使用其他环境文件将环境文件添加到堆栈中，并部署 overcloud。

2.9. 块存储服务(CINDER)一致性组

您可以使用 Block Storage (cinder)服务将一致性组设置为将多个卷作为单一实体分组在一起。这意味着，您可以同时对多个卷执行操作，而不是单独执行。您可以使用一致性组为多个卷同时创建快照。这也意味着您可以同时恢复或克隆这些卷。

卷可以是多个一致性组的成员。但是，在将卷添加到一致性组后，您无法删除、重新输入或迁移卷。

2.9.1. 配置块存储服务一致性组

默认情况下，块存储安全策略禁用一致性组 API。在使用该功能前，您必须在此处启用它。托管 Block Storage API 服务的节点 **/etc/cinder/policy.json** 文件中的相关一致性组条目， **openstack-cinder-api** 列出默认设置：

```
"consistencygroup:create" : "group:nobody",
"consistencygroup:delete": "group:nobody",
"consistencygroup:update": "group:nobody",
"consistencygroup:get": "group:nobody",
"consistencygroup:get_all": "group:nobody",
"consistencygroup:create_cgsnapshot" : "group:nobody",
"consistencygroup:delete_cgsnapshot": "group:nobody",
"consistencygroup:get_cgsnapshot": "group:nobody",
"consistencygroup:get_all_cgsnapshots": "group:nobody",
```

您必须在环境文件中更改这些设置，然后使用 **openstack overcloud deploy** 命令将它们部署到 overcloud。不要直接编辑 JSON 文件，因为在下次部署 overcloud 时更改会被覆盖。

先决条件

- 已安装 undercloud。如需更多信息，请参阅使用 director *安装和管理 Red Hat OpenStack Platform 中的安装 director*。

流程

1. 以 **stack** 用户身份登录 undercloud 主机。
2. 查找 **stackrc** undercloud 凭证文件：

```
$ source ~/stackrc
```

3. 编辑环境文件，并在 **parameter_defaults** 部分中添加新条目。这样可确保条目在容器中更新，并在 director 使用 **openstack overcloud deploy** 命令重新部署环境时保留条目。

4. 使用 **CinderApiPolicies** 在环境文件中添加新部分来设置一致性组设置。带有 JSON 文件中的默认设置的等同 **parameter_defaults** 部分如下所示：

```
parameter_defaults:
  CinderApiPolicies: { \
     cinder-consistencygroup_create: { key: 'consistencygroup:create', value: 'group:nobody' },
\
     cinder-consistencygroup_delete: { key: 'consistencygroup:delete', value: 'group:nobody' },
\
     cinder-consistencygroup_update: { key: 'consistencygroup:update', value: 'group:nobody'
}, \
     cinder-consistencygroup_get: { key: 'consistencygroup:get', value: 'group:nobody' }, \
     cinder-consistencygroup_get_all: { key: 'consistencygroup:get_all', value: 'group:nobody' },
\
     cinder-consistencygroup_create_cgsnapshot: { key:
'consistencygroup:create_cgsnapshot', value: 'group:nobody' }, \
     cinder-consistencygroup_delete_cgsnapshot: { key:
'consistencygroup:delete_cgsnapshot', value: 'group:nobody' }, \
     cinder-consistencygroup_get_cgsnapshot: { key: 'consistencygroup:get_cgsnapshot',
value: 'group:nobody' }, \
     cinder-consistencygroup_get_all_cgsnapshots: { key:
'consistencygroup:get_all_cgsnapshots', value: 'group:nobody' }, \
  }
```

5. 值 **'group:nobody'** 确定没有组可以使用此功能，以便有效地禁用它。要启用它，请将组更改为另一个值。

6. 为提高安全性，请将一致性组 API 和卷类型管理 API 的权限设置为相同。默认情况下，卷类型管理 API 会在同一个 **/etc/cinder/policy.json_ file** 文件中设置为 **"rule:admin_or_owner"**。

```
"volume_extension:types_manage": "rule:admin_or_owner",
```

7. 要使一致性组功能可供所有用户使用，请将 API 策略条目设置为允许用户创建、使用和管理自己的一致性组。要做到这一点，请使用 **rule:admin_or_owner**:

```
CinderApiPolicies: { \
     cinder-consistencygroup_create: { key: 'consistencygroup:create', value:
'rule:admin_or_owner' }, \
     cinder-consistencygroup_delete: { key: 'consistencygroup:delete', value:
'rule:admin_or_owner' },  \
     cinder-consistencygroup_update: { key: 'consistencygroup:update', value:
'rule:admin_or_owner' }, \
     cinder-consistencygroup_get: { key: 'consistencygroup:get', value: 'rule:admin_or_owner'
}, \
     cinder-consistencygroup_get_all: { key: 'consistencygroup:get_all', value:
'rule:admin_or_owner' }, \
     cinder-consistencygroup_create_cgsnapshot: { key:
'consistencygroup:create_cgsnapshot', value: 'rule:admin_or_owner' }, \
     cinder-consistencygroup_delete_cgsnapshot: { key:
'consistencygroup:delete_cgsnapshot', value: 'rule:admin_or_owner' }, \
     cinder-consistencygroup_get_cgsnapshot: { key: 'consistencygroup:get_cgsnapshot',
value: 'rule:admin_or_owner' }, \
     cinder-consistencygroup_get_all_cgsnapshots: { key:
'consistencygroup:get_all_cgsnapshots', value: 'rule:admin_or_owner’ }, \
  }
```

8. 将更新保存到环境文件中。
9. 使用其他环境文件将环境文件添加到堆栈中，并部署 overcloud。

2.9.2. 使用仪表板创建块存储一致性组

启用一致性组 API 后，您可以开始创建一致性组。

先决条件

- 您必须是项目管理员或卷所有者才能创建一致性组。
- 访问 Red Hat OpenStack Platform (RHOSP) Dashboard (horizon)。如需更多信息，请参阅 *使用 OpenStack 控制面板管理云资源*。

流程

1. 以 admin 用户或卷所有者登录控制面板。
2. 选择 **Project > Compute > Volumes > Volume Consistency Groups**
3. 单击 **Create Consistency Group**。
4. 在向导的 **Consistency Group Information** 选项卡中，输入您的一致性组的名称和描述。然后，指定其 **可用区**。
5. 您还可以将卷类型添加到一致性组中。当您在一致性组中创建卷时，块存储服务将从这些卷类型应用兼容的设置。要添加卷类型，请点击**所有可用卷类型**列表中的 **+** 按钮。
6. 单击 **Create Consistency Group**。它会出现在 **Volume Consistency Groups** 表中。

2.9.3. 使用控制面板管理块存储服务一致性组

您可以在仪表板中管理块存储卷的一致性组。

先决条件

- 您必须是项目管理员来管理一致性组。
- 访问 Red Hat OpenStack Platform (RHOSP) Dashboard (horizon)。如需更多信息，请参阅 *使用 OpenStack 控制面板管理云资源*。

流程

1. 以 admin 用户身份登录控制面板。
2. 选择 **Project > Compute > Volumes > Volume Consistency Groups**
3. 可选：您可以通过从 **Action** 列中选择 **Edit Consistency Group** 来更改一致性组的名称或描述。
4. 要直接从一致性组中添加或删除卷，请找到您要配置的一致性组。在该一致性组的 **Actions** 列中，选择 **Manage Volumes**。这将启动 **Add/Remove Consistency Group Volumes** 向导。
   a. 要将卷添加到一致性组中，请单击 **All available volumes** 列表中的 **+** 按钮。
   b. 要从一致性组中删除卷，请单击 **Selected volumes** 列表中的 **-** 按钮。

5. 单击 **Edit Consistency Group**。

2.9.4. 为块存储服务创建和管理一致性组快照

将卷添加到一致性组后，您可以从其中创建快照。

先决条件

- 您必须是一个项目管理员，才能创建和管理一致性组快照。

流程

1. 提供 overcloud 凭证文件：

```
$ source ~/<credentials_file>
```

   - 将 **<credentials_file>** 替换为您的凭据文件的名称，如 **overcloudrc**。

2. 列出所有可用的一致性组及其对应 ID：

```
$ cinder consisgroup-list
```

3. 使用一致性组创建快照：

```
$ cinder cgsnapshot-create [--name <cgsnapname>] [--description "<description>"]
<cgnameid>
```

   - 将 **<cgsnapname>** 替换为快照的名称。
   - 将 **<description>** 替换为快照的描述。
   - 将 **<cgnameid>** 替换为一致性组的名称或 ID。

4. 显示所有可用一致性组快照的列表：

```
# cinder cgsnapshot-list
```

2.9.5. 克隆块存储服务一致性组

您还可以使用一致性组同时创建整个预配置的卷。您可以通过克隆现有的一致性组或恢复一致性组快照来完成此操作。两个进程都使用相同的命令。

先决条件

- 您必须是项目管理员，才能克隆一致性组和恢复一致性组快照。

流程

1. 提供 overcloud 凭证文件：

```
$ source ~/<credentials_file>
```

   - 将 **<credentials_file>** 替换为您的凭据文件的名称，如 **overcloudrc**。

2. 克隆现有的一致性组：

```
$ cinder consisgroup-create-from-src --source-cg <cgnameid> [--name <cgname>] [--description "<description>"]
```

- 将 **<cgnameid** > 替换为您要克隆的一致性组的名称或 ID。
- 将 **<cgname** > 替换为您的一致性组的名称。
- 将 **<description** > 替换为您的一致性组的描述。

3. 从一致性组快照创建一致性组：

```
$ cinder consisgroup-create-from-src --cgsnapshot <cgsnapname> [--name <cgname>] [--description "<description>"]
```

- 将 < **cgsnapname** > 替换为您要用于创建一致性组的快照的名称或 ID。

2.10. 配置默认块存储调度程序过滤器

如果在卷创建过程中没有指定卷后端，则块存储调度程序将使用过滤器来选择合适的后端。确保配置以下默认过滤器：

AvailabilityZoneFilter

过滤掉不符合所请求卷的可用区要求的所有后端。

CapacityFilter

仅选择具有足够空间的后端来容纳卷。

CapabilitiesFilter

仅选择可支持卷中任何指定设置的后端。

InstanceLocality

将集群配置为使用卷本地到同一节点。

先决条件

- 已安装 undercloud。如需更多信息，请参阅使用 director *安装和管理 Red Hat OpenStack Platform 中的安装 director*。

流程

1. 以 **stack** 用户身份登录 undercloud 主机。
2. 查找 **stackrc** undercloud 凭证文件：

```
$ source ~/stackrc
```

3. 在 overcloud 部署命令中添加一个环境文件，其中包含以下参数：

```
parameter_defaults:
  ControllerExtraConfig: # (1)
    cinder::config::cinder_config:
      DEFAULT/scheduler_default_filters:
        value: 'AvailabilityZoneFilter,CapacityFilter,CapabilitiesFilter,InstanceLocality'
```

-

1 您还可以在现有环境文件的 **parameter_defaults:** 部分中添加 **ControllerExtraConfig:** hook 及其嵌套部分。

4. 将更新保存到环境文件中。

5. 使用其他环境文件将环境文件添加到堆栈中，并部署 overcloud。

2.11. 在 OVERCLOUD 节点上启用 LVM2 过滤

如果您使用带有某些块存储服务(cinder)后端的 LVM2 （逻辑卷管理）卷，您在 Red Hat OpenStack Platform (RHOSP)客户机中创建的卷可能会在主机 **cinder-volume** 或 **nova-compute** 容器的 overcloud 节点上可见。在这种情况下，主机上的 LVM2 工具扫描 OpenStack 客户机创建的 LVM2 卷，这可能会在 Compute 或 Controller 节点上产生一个或多个问题：

- LVM 似乎查看客户机中的卷组
- LVM 报告重复的卷组名称
- 卷分离失败，因为 LVM 正在访问存储
- 由于 LVM 存在问题，客户机无法引导
- 因为缺少实际存在的磁盘，客户机机器上的 LVM 处于部分状态
- 在具有 LVM 的设备中，块存储服务(cinder)操作会失败
- 块存储服务(cinder)快照无法正确删除
- 实时迁移过程中出现错误： **/etc/multipath.conf** 不存在

要防止这种错误扫描，并从主机节点隔离客户机 LVM2 卷，您可以在部署或更新 overcloud 时使用 **LVMFilterEnabled** heat 参数启用和配置过滤器。这个过滤器从托管活跃 LVM2 卷的物理设备列表中计算。您还可以使用 **LVMFilterAllowlist** 和 **LVMFilterDenylist** 参数明确允许或拒绝块设备。您可以在全局范围内、针对特定节点角色或特定设备应用此过滤。

先决条件

- 已安装 undercloud。如需更多信息，请参阅使用 director *安装和管理 Red Hat OpenStack Platform 中的安装 director*。

流程

1. 以 **stack** 用户身份登录 undercloud 主机。
2. 查找 **stackrc** undercloud 凭证文件：

```
$ source ~/stackrc
```

3. 在 overcloud 部署命令中添加一个环境文件，其中包含以下参数：

```
parameter_defaults:
  LVMFilterEnabled: true
```

您可以进一步自定义 LVM2 过滤器的实现。例如，要只在 Compute 节点上启用过滤，请使用以下配置：

```
parameter_defaults:
  ComputeParameters:
    LVMFilterEnabled: true
```

这些参数还支持正则表达式。要只在 Compute 节点上启用过滤，并忽略所有以 **/dev/sd** 开头的设备，请使用以下配置：

```
parameter_defaults:
  ComputeParameters:
    LVMFilterEnabled: true
    LVMFilterDenylist:
      - /dev/sd.*
```

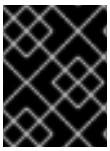
4. 将更新保存到环境文件中。
5. 使用其他环境文件将环境文件添加到堆栈中，并部署 overcloud。

2.12. 多路径配置

使用多路径将服务器节点和存储阵列间的多个 I/O 路径配置为单一设备，以创建冗余并提高性能。

2.12.1. 使用 director 配置多路径

您可以在 Red Hat OpenStack Platform (RHOSP) overcloud 部署中配置多路径，以实现更大的带宽和网络弹性。



重要

当您在现有部署中配置多路径时，新的工作负载会识别多路径。如果您有任何已存在的工作负载，您必须检查并取消缩小实例以便在这些实例上启用多路径。

先决条件

- 已安装 undercloud。如需更多信息，请参阅使用 director *安装和管理 Red Hat OpenStack Platform 中的安装 director*。

流程

1. 以 **stack** 用户身份登录 undercloud 主机。
2. 查找 **stackrc** undercloud 凭证文件：

```
$ source ~/stackrc
```

3. 使用覆盖环境文件或创建新文件，如 **multipath_overrides.yaml**。添加并设置以下参数：

```
parameter_defaults:
  ExtraConfig:
    cinder::config::cinder_config:
```

```
backend_defaults/use_multipath_for_image_xfer:
  value: true
```

注意

默认设置将生成适用于大多数环境的基本多路径配置。但是，检查您的存储供应商以获得建议，因为有些供应商有特定于硬件的优化配置。有关多路径的更多信息，*请参阅配置设备映射器多路径*。

4. 可选：如果您有用于 overcloud 部署的多路径配置文件，您可以使用 **MultipathdCustomConfigFile** 参数指定此文件的位置：

```
parameter_defaults:
  MultipathdCustomConfigFile: <config_file_directory>/<config_file_name>
```

在以下示例中，**/home/stack** 是多路径配置文件的目录，**multipath.conf** 是此文件的名称：

```
parameter_defaults:
  MultipathdCustomConfigFile: /home/stack/multipath.conf
```

注意

其他 TripleO 多路径参数会覆盖本地自定义配置文件中的任何对应的值。例如，如果 **MultipathdEnableUserFriendlyNames** 是 **False**，则 overcloud 节点上的文件会更新以匹配，即使本地自定义文件中启用了该设置。

有关多路径参数的更多信息，请参阅 多路径 heat 模板参数。

5. 将更新保存到覆盖的环境文件。
6. 将覆盖环境文件添加到带有其他环境文件的堆栈中，例如：

```
----
/usr/share/openstack-tripleo-heat-templates/environments/multipathd.yaml
----
```

7. 部署 overcloud。

其他资源

- *在 创建和管理实例中分离实例*

2.12.1.1. 多路径 heat 模板参数

使用它来了解启用多路径的以下参数。

参数	描述	默认值
MultipathdEnable	定义是否启用多路径守护进程。通过 **multipathd.yaml** 文件中包含的配置，此参数默认为 **True**	**true**

参数	描述	默认值
MultipathdEnableUserFriendlyNames	定义是否启用为每个路径分配用户友好的名称。	**False**
MultipathdEnableFindMultipaths	定义是否为每个路径自动创建多路径设备。	**true**
MultipathdSkipKpartx	定义是否跳过设备中自动创建分区。	**true**
MultipathdCustomConfigFile	包括 overcloud 节点上的本地自定义多路径配置文件。默认情况下会安装一个最小 **multipath.conf** 文件。 **注意：** 其他 TripleO 多路径参数覆盖您添加的任何本地自定义配置文件中任何对应的值。例如，如果 **MultipathdEnableUserFriendlyNames** 为 **False**，则 overcloud 节点上的文件会更新以匹配，即使在本地、自定义文件中启用了该设置。	

2.12.2. 验证多路径配置

您可以在新的或现有 overcloud 部署中验证多路径配置。

流程

1. 创建一个实例。
2. 将非加密卷附加到实例。
3. 获取包含实例的 Compute 节点的名称：

```
$ nova show <instance> | grep OS-EXT-SRV-ATTR:host
```

将 **<instance** > 替换为您创建的实例的名称。

4. 检索实例的 virsh 名称：

```
$ nova show <instance> | grep instance_name
```

5. 获取 Compute 节点的 IP 地址：

```
$ . stackrc
$ metalsmith list | grep <compute_name>
```

将 **<compute_name** > 替换为 **nova show <instance** > 命令的输出中的 name，从六个列表中显示两行。

找到在第四列中的 & **lt;compute_name** > 的行。< **compute_name&gt**; 的 IP 地址位于此行的最后一列中。

在以下示例中，compute-0 的 IP 地址是 192.02.24.15，因为 compute-0 位于第二行的第四列中：

```
$ . stackrc
$ metalsmith list | grep compute-0
| 3b1bf72e-c425-494c-9717-d0b89bb66580 | compute-0    | 95b21d3e-36be-470d-ba5c-70d5dcd6d0b3 | compute-1    | ACTIVE | ctlplane=192.02.24.49 |
| 72a24883-25f9-435c-bf71-a20e66be172d | compute-1    | a59f79f7-006e-4f38-a9ad-8164da47d58e | compute-0    | ACTIVE | ctlplane=192.02.24.15 |
```

6. SSH 到运行实例的 Compute 节点：

```
$ ssh tripleo-admin@<compute_node_ip>
```

将 **<compute_node_ip** > 替换为 Compute 节点的 IP 地址。

7. 登录到运行 virsh 的容器：

```
$ podman exec -it nova_libvirt /bin/bash
```

8. 在 Compute 节点实例中输入以下命令，以验证它是否在 cinder 卷主机位置中使用多路径：

```
virsh domblklist <virsh_instance_name> | grep /dev/dm
```

将 **<virsh_instance_name** > 替换为 **nova show <instance> | grep instance_name** 命令的输出。

如果实例显示 **/dev/dm-** 以外的值，则连接是非多路径，您需要使用 **nova shelve** 和 **nova unshelve** 命令刷新连接信息：

```
$ nova shelve <instance>
$ nova unshelve <instance>
```

注意

如果您有多种后端，您必须验证所有后端上的实例和卷，因为每个后端返回的连接信息可能有所不同。

第 3 章 使用块存储服务(CINDER)执行基本操作

在您的 overcloud 中创建和配置块存储卷，作为计算实例的主要形式的持久性存储。创建卷，将卷附加到实例，编辑并调整卷所有权，以及修改卷所有权。

3.1. 创建块存储卷

创建卷，为您使用 overcloud 中的计算服务(nova)启动的实例提供持久存储。

要创建加密卷，您必须首先为卷加密配置了卷类型。另外，您必须配置 Compute 和 Block Storage 服务，以使用相同的静态密钥。有关如何设置卷加密要求的详情，请参考 Block Storage 服务(cinder)卷加密。

重要

您可以为项目创建的默认最大卷数量为 10。

先决条件

- 访问 Red Hat OpenStack Platform (RHOSP) Dashboard (horizon)。如需更多信息，请参阅 *使用 OpenStack 控制面板管理云资源*。

流程

1. 登录控制面板。
2. 选择 **Project > Compute > Volumes**。
3. 点 **Create Volume**，并编辑以下字段：

字段	描述
卷名称	卷的名称。
描述	可选，卷的简短描述。
类型	可选卷类型。如需更多信息，请参阅使用 卷类型分组卷配置。 如果您创建卷且没有指定卷类型，则块存储将使用默认卷类型。有关定义默认卷类型的更多信息，请参阅 定义特定于项目的默认卷类型。 如果没有指定后端，则块存储调度程序将尝试为您选择合适的后端。如需更多信息，请参阅 多个后端上的卷分配。 **注意** 如果没有合适的后端，则不会创建卷。 您还可以在卷创建后更改卷类型。如需更多信息，请参阅 块存储卷调整。

字段	描述
Size (GB)	卷大小（以 GB 为单位）。 如果要从未加密的镜像创建加密卷，您必须确保卷大小大于镜像大小，以便加密数据不会截断卷数据。
可用性区域	可用性区域（逻辑服务器组）以及主机聚合是隔离 OpenStack 中资源的常用方法。可用区在安装过程中定义。如需有关可用区和主机聚合的更多信息，请参阅配置 *计算服务以进行实例创建* 指南中的 创建和管理主机聚合。

4. 指定 **卷源** ：

源	描述
没有源的空卷	卷为空，不包含它 文件系统或分区表。
Snapshot	使用现有快照作为卷源。如果选择了这个选项，则会打开一个新的 **Use snapshot 作为** 源列表；您可以从列表中选择一个快照。如果要从加密卷快照创建新卷，您必须确保新卷至少大于旧卷。有关卷快照的更多信息，请参阅从快照创建新卷。
Image	使用现有镜像作为卷源。如果选择了这个选项，则会打开一个新的 **Use snapshot 作为** 源列表；您可以从列表中选择一个镜像。
卷	使用现有卷作为卷源。如果选择了这个选项，则会打开一个新的 **Use snapshot 作为** 源列表；您可以从列表中选择一个卷。

5. 点**创建卷**。创建卷后，其名称将显示在 **Volumes** 表中。

3.2. 编辑卷名称或描述

您可以在仪表板中更改卷的名称和描述。

先决条件

- 访问 Red Hat OpenStack Platform (RHOSP) Dashboard (horizon)。如需更多信息，请参阅 *使用 OpenStack 控制面板管理云资源*。

流程

1. 登录控制面板。

2. 选择 **Project > Compute > Volumes**。
3. 选择卷的 **编辑卷** 按钮。
4. 根据需要编辑卷名称或描述。
5. 点 **Edit Volume** 保存您的更改。

3.3. 重新定义块存储服务卷大小（扩展）

重新定义卷大小以增加卷的存储容量。

注意

支持调整使用中的卷大小，但取决于驱动程序。RBD 支持。您不能扩展使用多重附加卷。有关支持这个功能的更多信息，请联系红帽支持。

流程

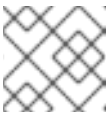
1. 提供您的凭据文件。
2. 列出卷以检索您要扩展的卷 ID：

```
$ cinder list
```

3. 增加卷的大小：

```
$ cinder extend <volume_id> <size>
```

- 将 **<volume_id>** 替换为您要扩展的卷的 ID。
- 将 **<size>** 替换为这个卷所需的大小，以 GB 为单位。



注意

确保指定的大小大于这个卷的现有大小。

例如：

```
$ cinder extend 573e024d-5235-49ce-8332-be1576d323f8 10
```

3.4. 删除块存储服务卷

您可以删除不再需要的卷。

注意

如果卷有现有的快照，则无法删除卷。有关删除快照的更多信息，请参阅 删除卷快照。

先决条件

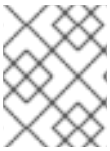
- 访问 Red Hat OpenStack Platform (RHOSP) Dashboard (horizon)。如需更多信息，请参阅 *使用 OpenStack 控制面板管理云资源*。

流程

1. 登录控制面板。
2. 选择 **Project > Compute > Volumes**。
3. 在 **Volumes** 表中，选择要删除的卷。
4. 点**删除卷**。

3.5. 在多个后端上的卷分配

当您创建卷时，您可以从 Type 列表中选择所需后端的卷类型。如需更多信息，请参阅创建块存储卷。



注意

如果将 Block Storage 服务(cinder)配置为使用多个后端，则必须为每个后端创建一个卷类型。

如果您在创建卷时没有指定后端，则块存储调度程序将尝试为您选择合适的后端。

调度程序为卷的默认关联设置使用过滤器来选择合适的后端：

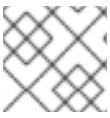
AvailabilityZoneFilter
过滤掉不符合所请求卷的可用区要求的所有后端。

CapacityFilter
仅选择具有足够空间的后端来容纳卷。

CapabilitiesFilter
仅选择可支持卷中任何指定设置的后端。

InstanceLocality
将集群配置为使用卷本地到同一节点。

如果有多个合适的后端，则调度程序使用权重方法选择最佳后端。默认情况下，使用 CapacityWeigher 方法，以便选择具有最多可用空间的过滤后端。



注意

如果没有合适的后端，则不会创建卷。

其他资源

- 创建和配置卷类型
- 块存储卷重新调整
- 配置默认块存储调度程序过滤器

3.6. 将卷附加到实例

当您关闭一个实例时，所有数据都会丢失。您可以为持久性存储附加卷。您可以一次将卷附加到一个实例，除非它有一个 multi-attach 卷类型。有关创建多附加卷的更多信息，请参阅可以附加到多个实例 的卷。

先决条件

- 访问 Red Hat OpenStack Platform (RHOSP) Dashboard (horizon)。如需更多信息，请参阅 *使用 OpenStack 控制面板管理云资源*。

流程

1. 登录控制面板。
2. 选择 **Project > Compute > Volumes**。
3. 选择 **Edit Attachments** 操作。如果卷没有附加到实例，则可以看到 **Attach To Instance** 下拉列表。
4. 从 **Attach To Instance** 列表中，选择要将卷附加到的实例。
5. 单击 **Attach Volume**。

3.7. 从实例分离卷

如果要将这个卷附加到另一个实例时，您必须将卷从实例分离，除非它有一个 multi-attach 卷类型。您还必须分离卷以更改卷的访问权限或删除卷。

先决条件

- 访问 Red Hat OpenStack Platform (RHOSP) Dashboard (horizon)。如需更多信息，请参阅 *使用 OpenStack 控制面板管理云资源*。

流程

1. 登录控制面板。
2. 选择 **Project > Compute > Volumes**。
3. 选择卷的 **Manage Attachments** 操作。如果卷附加到实例，实例的名称将显示在 **Attachments** 表中。
4. 单击此中的 **Detach Volume**，然后单击下一个对话框屏幕。

后续步骤

- 将卷附加到实例

3.8. 配置卷的访问权限

卷的默认状态是读写的，以允许写入数据并从中读取数据。您可以将卷标记为只读，以防止其数据被意外覆盖或删除。

注意

将卷更改为只读后，您可以再次将其更改回读写。

先决条件

- 如果卷已附加到实例，则分离这个卷。如需更多信息，请参阅从实例分离卷。

流程

1. 提供您的凭据文件。
2. 列出卷以检索您要配置的卷的 ID：

```
$ cinder list
```

3. 设置此卷所需的访问权限：
   - 将卷的访问权限设置为只读：

```
$ cinder readonly-mode-update <volume_id> true
```

     - 将 **<volume_id** > 替换为所需卷的 ID。
   - 将卷的访问权限设置为读写：

```
$ cinder readonly-mode-update <volume_id> false
```

4. 如果您将这个卷从实例分离以更改访问权限，则重新附加卷。如需更多信息，请参阅 将卷附加到实例。

3.9. 使用仪表板更改卷所有者

要更改卷的所有者，您必须执行卷转让。卷转让由卷的所有者启动，在卷的新所有者接受转让后，卷的所有权更改已完成。

先决条件

- 访问 Red Hat OpenStack Platform (RHOSP) Dashboard (horizon)。如需更多信息，请参阅 *使用 OpenStack 控制面板管理云资源*。

流程

1. 以卷所有者身份登录控制面板。
2. 选择 **Projects > Volumes**。
3. 在要转让的卷的 **Actions** 列中，选择 **Create Transfer**。
4. 在 **Create Transfer** 对话框中，输入转让的名称，再单击 **Create Volume Transfer**。
   卷转让已创建，在 **Volume Transfer** 屏幕中，您可以捕获发送到接收者项目的 **转让 ID 和授权密钥**。

点 Download transfer credentials 按钮下载一个 **.txt** 文件，它包括了 **transfer name**, **transfer ID**, 和 **authorization key**。

注意

授权密钥仅在 Volume Transfer 屏幕中提供。如果丢失了授权密钥，您必须取消转让并创建另一个传输来生成新的授权密钥。

5. 关闭 Volume Transfer 屏幕，以返回到卷列表。
卷状态会变为 **awaiting-transfer**，直到接收者项目接受转让为止

从仪表板接受卷转让

1. 以接收者项目所有者登录控制面板。
2. 选择 **Projects > Volumes**。
3. 单击 **Accept Transfer**。
4. 在 **Accept Volume Transfer** 对话框中，输入您从卷所有者接收的 **转让 ID 和授权密钥**，然后单击 **Accept Volume Transfer**。
卷现在出现在活动项目的卷列表中。

3.10. 使用 CLI 更改卷所有者

要更改卷的所有者，您必须执行卷转让。卷转让由卷的所有者启动，在卷的新所有者接受转让后，卷的所有权更改已完成。

流程

1. 以卷的当前所有者身份登录。
2. 列出可用的卷：

```
$ cinder list
```

3. 启动卷转让：

```
$ cinder transfer-create <volume>
```

将 **<volume >** 替换为您要传输的卷的名称或 ID。例如：

```
+------------+--------------------------------------+
|  Property  |                Value                 |
+------------+--------------------------------------+
|  auth_key  |           f03bf51ce7ead189          |
| created_at |      2014-12-08T03:46:31.884066      |
|     id     | 3f5dc551-c675-4205-a13a-d30f88527490 |
|    name    |                 None                 |
| volume_id  | bcf7d015-4843-464c-880d-7376851ca728 |
+------------+--------------------------------------+
```

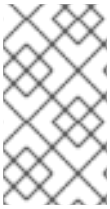
cinder transfer-create 命令清除卷的所有权，并为转让创建一个 **id** 和 **auth_key**。这些值可以被指定为，供另一个用户用来接受转让，并成为卷的新所有者。

4. 新用户现在可以声明卷的所有权。要做到这一点，用户应首先从命令行登录并运行：

```
$ cinder transfer-accept <transfer_id> <transfer_key>
```

- 将 **<transfer_id** > 替换为 **cinder transfer-create** 命令返回的 **id** 值。
- 将 **<transfer_key** > 替换为 **cinder transfer-create** 命令返回的 **auth_key** 值。
  例如：

```
$ cinder transfer-accept 3f5dc551-c675-4205-a13a-d30f88527490 f03bf51ce7ead189
```



注意

您可以使用以下方法查看所有可用卷传输：

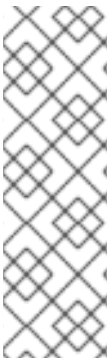
```
$ cinder transfer-list
```

第 4 章 使用块存储服务(CINDER)执行高级操作.

块存储卷为您的 overcloud 中的计算实例组成持久性存储。配置卷的高级功能，例如，使用卷快照，创建多附加卷、重新调整卷和迁移卷。

4.1. 创建卷快照

您可以通过创建卷快照来保留卷在特定时间点的状态。然后，您可以使用快照克隆新卷。



注意

卷备份与快照不同。备份保留卷中包含的数据，而快照会在特定时间点保留卷的状态。如果卷有现有的快照，则无法删除卷。卷备份可防止数据丢失，而快照则有助于克隆。

因此，快照后端通常与卷后端在一起，以便在克隆过程中最小化延迟。相反，备份存储库通常位于不同的位置，例如在典型的企业部署中在不同节点、物理存储甚至地理位置。这是为了防止备份存储库不受卷后端可能出现的任何损坏的影响。

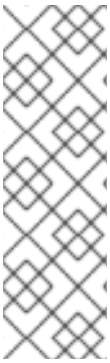
有关卷备份的更多信息，请参阅 *备份块存储卷* 指南。

先决条件

- 要快照的卷。有关创建卷的更多信息，请参阅创建块存储卷。
- 访问 Red Hat OpenStack Platform (RHOSP) Dashboard (horizon)。如需更多信息，请参阅 *使用 OpenStack 控制面板管理云资源*。

流程

1. 登录控制面板。
2. 选择 **Project > Compute > Volumes**。
3. 为目标卷选择 **Create Snapshot** 操作。
4. 为 **快照** 提供快照名称，再单击 **创建卷快照**。**卷快照** 选项卡显示所有快照。



注意

对于从快照创建的块存储服务(cinder)的 RADOS 块设备(RBD)卷，您可以使用 **CinderRbdFlattenVolumeFromSnapshot** heat 参数扁平化和删除对快照的依赖项。当您将 **CinderRbdFlattenVolumeFromSnapshot** 设置为 **true** 时，块存储服务扁平化 RBD 卷，并删除对快照的依赖项，同时扁平化所有未来的快照。默认值为 **false**，这也是 cinder RBD 驱动程序的默认值。

请注意，扁平化快照会删除与父级共享任何潜在的块，并在后端上产生更大的快照大小，并增加快照创建时间。

验证

- 验证新快照出现在 **Volume Snapshots** 标签页中，或使用 CLI 列出卷快照，并验证快照是否已创建：

```
$ openstack volume snapshot list
```

4.2. 从快照创建新卷

您可以创建新卷作为卷快照的克隆。这些快照会在特定时间点保留卷的状态。

先决条件

- 要从中克隆并创建新卷的卷快照。有关创建卷快照的更多信息，请参阅 创建卷快照
- 访问 Red Hat OpenStack Platform (RHOSP) Dashboard (horizon)。如需更多信息，请参阅 *使用 OpenStack 控制面板管理云资源*。

流程

1. 登录控制面板。
2. 选择 **Project > Compute > Volumes**。
3. 在 **Volume Snapshots** 表中，为您要从中创建新卷的快照选择 **Create Volume** 操作。有关卷创建的更多信息，请参阅创建块存储卷。

重要

如果要从加密卷快照创建新卷，您必须确保新卷至少大于旧卷。

验证

- 验证新卷出现在 **Volumes** 标签页中，或使用 CLI 列出卷并验证已创建了新卷：

```
$ openstack volume list
```

4.3. 删除卷快照

Red Hat OpenStack Platform (RHOSP) 17.1 使用 RBD CloneV2 API，这意味着您可以删除卷快照，即使它们有依赖项。如果您的 RHOSP 部署使用 director 部署的 Ceph 后端，则 director 会正确配置 Ceph 集群。

如果使用外部 Ceph 后端，您必须在 Ceph 集群上配置最小客户端。有关配置外部 Ceph 集群的更多信息，请参阅 *Integrating the overcloud with an existing Red Hat Ceph Storage Cluster* 中的配置现有的 Red Hat Ceph Storage 集群。

先决条件

- 要删除的卷快照。
- 访问 Red Hat OpenStack Platform (RHOSP) Dashboard (horizon)。如需更多信息，请参阅 *使用 OpenStack 控制面板管理云资源*。

流程

1. 登录控制面板。
2. 选择 **Project > Compute > Volumes**。
3. 在 **Volume Snapshots** 表中，为您要删除的快照选择 **Delete Volume Snapshot** 操作。

如果您的 OpenStack 部署使用 Red Hat Ceph 后端，以了解有关快照安全和故障排除的更多信息，请参阅 Red Hat Ceph Storage 后端中的保护和未保护的快照 。

验证

- 验证快照是否不再存在在 **Volume Snapshots** 选项卡中，或使用 CLI 列出卷快照并验证快照是否已删除：

```
$ openstack volume snapshot list
```

4.4. 从快照中恢复卷

您可以通过将卷数据恢复到其最新快照来恢复卷的最新快照。

警告

支持恢复卷的最新快照，但取决于驱动程序。有关支持这个功能的更多信息，请联系您的驱动程序厂商。

限制

- 在多附加卷中使用恢复到快照功能可能会有限制。在使用此功能前，检查是否应用了这些限制。
- 在进行快照后，您无法恢复调整（扩展）的卷。
- 您不能在附加或正在使用的卷中使用 restore-to-snapshot 功能。
- 默认情况下，您无法在可引导的根卷中使用 restore-to-snapshot 功能。要使用这个功能，您必须使用 **delete_on_termination=false** 属性引导实例，以便在实例终止时保留引导卷。在这种情况下，要恢复到快照，您必须：
  - 删除实例，使卷可用，然后
  - 恢复卷，然后恢复卷
  - 从卷创建一个新实例。

先决条件

- 块存储(cinder) REST API 微版本 3.40 或更高版本。
- 您必须至少为卷创建了一个快照。

流程

1. 提供您的凭据文件。
2. 分离卷：

```
$ openstack server remove volume <instance_id> <vol_id>
```

- 将 **<instance_id > 和 <vol_id** > 替换为您要恢复的实例和卷的 ID。

3. 找到您要恢复的快照的 ID 或名称。您只能恢复最新的快照。

```
$ cinder snapshot-list
```

4. 恢复快照：

```
$ cinder --os-volume-api-version=3.40 revert-to-snapshot <snapshot_id>
```

- 将 **<snapshot_id** > 替换为快照的 ID。

5. 可选： 您可以使用 **cinder snapshot-list** 命令检查您要恢复的卷是否处于恢复状态。

```
$  cinder snapshot-list
```

6. 重新附加卷：

```
$ openstack server add volume <instance_id> <vol_id>
```

- 将 **<instance_id** & **gt; 和 <vol_id** > 替换为您恢复的实例和卷的 ID。

验证

- 要检查流程是否成功，您可以使用 **cinder list** 命令验证您恢复的卷现在是否处于 available 状态。

```
$ cinder list
```

4.5. 上传卷到镜像服务(GLANCE)

您可以将现有卷作为镜像直接上传到镜像服务。

先决条件

- 访问 Red Hat OpenStack Platform (RHOSP) Dashboard (horizon)。如需更多信息，请参阅 *使用 OpenStack 控制面板管理云资源*。

流程

1. 登录控制面板。
2. 选择 **Project > Compute > Volumes**。
3. 选择目标卷的 **Upload to Image** 操作。
4. 为卷提供 **Image Name**，并从列表中选择 **Disk Format**。
5. 点 **Upload**。

要查看上传的镜像，请选择 **Project > Compute > Images**。新镜像会出现在 **Images** 表中。有关如何使用和配置镜像的详情，请参考 *创建和管理镜像*。

4.6. 可附加到多个实例的卷

您可以创建一个多附件块存储卷，它可以附加到多个实例，并且这些实例可以同时读取和写入它。multi-attach 卷需要一个 multi-attach 卷类型。

警告

您必须使用多附件或集群感知的文件系统来管理来自多个实例的写入操作。如果不这样做，会导致数据崩溃。

多附加卷的限制

- Block Storage (cinder)后端必须支持多附加卷。有关支持的后端的详情，请联系红帽支持。
- 您的 Block Storage (cinder)驱动程序必须支持多附加卷。支持 Ceph RBD 驱动程序。联系红帽支持，以验证您的厂商插件是否支持 multi-attach。有关厂商插件认证的更多信息，请参阅以下位置：
  - https://access.redhat.com/articles/1535373#Cinder
  - https://access.redhat.com/ecosystem/search/#/category/Software?sort=sortTitle%20asc&softwareCategories=Storage&ecosystem=Red%20Hat%20OpenStack%
- 不支持只读多附件卷。
- 多附加卷的实时迁移不可用。
- 不支持对多重附加卷的加密。
- Bare Metal Provisioning 服务(ironic) virt 驱动程序不支持 multi-attach 卷。只有在 libvirt virt 驱动程序才支持 multi-attach 卷。
- 您不能将附加的卷从 multi-attach 类型重新输入到非附加类型，您无法将非 multi-attach 类型重新输入到 multi-attach 类型。
- 在附加的卷迁移过程中，您无法使用多个读写附加作为源或目标卷的多附件。
- 您不能将多附件卷附加到 shelved 卸载的实例。

4.6.1. 创建多附加卷类型

要将卷附加到多个实例，请在卷额外规格中将 **multiattach** 标志设置为 & **lt;is> True**。当您创建 multi-attach 卷类型时，卷将继承 标记并成为多附件卷。

先决条件

- 您必须是项目管理员才能创建卷类型。

流程

1. 提供 overcloud 凭证文件：

```
$ source ~/<credentials_file>
```

- 将 **<credentials_file>** 替换为您的凭据文件的名称，如 **overcloudrc**。

2. 为 multi-attach 卷创建新卷类型：

```
$ cinder type-create multiattach
```

3. 为此 multi-attach 卷类型启用 **multiattach** 属性：

```
$ cinder type-key multiattach set multiattach="<is> True"
```

4. 运行以下命令来指定后端：

```
$ cinder type-key multiattach set volume_backend_name=<backend_name>
```

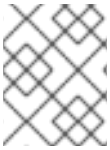
4.6.2. 多附加卷重新处理

您可以重新键入卷，使其能够多重附加，或者重新输入支持多重附加的卷，使其能够附加到多个实例。但是，您只能在不使用时重新键入卷，其状态为 **可用**。

当您附加一个多附件卷时，一些虚拟机监控程序需要特殊考虑，比如当您禁用缓存时。目前，无法安全地更新附加的卷，同时保留整个卷。如果您试图重新输入附加到多个实例的卷，则重新处理会失败。

4.6.3. 创建一个多附件卷

您可以创建一个可附加到多个实例的块存储卷，这些实例可以同时读取和写入它。



注意

此流程在支持 **multiattach** 的任何后端上创建卷。因此，如果支持 **multiattach** 的两个后端，调度程序会决定使用哪个后端。如需更多信息，请参阅 多个后端上的卷分配。

先决条件

- 您的项目中提供了多附件卷类型。

流程

1. 提供您的凭据文件。
2. 运行以下命令来创建 multi-attach 卷：

```
$ cinder create <volume_size> --name <volume_name> --volume-type multiattach
```

3. 运行以下命令，以验证卷是否能够多重附加。如果卷可以 multi-attach，则 **multiattach** 字段等于 **True**。

```
$ cinder show <vol_id> | grep multiattach

| multiattach | True |
```

后续步骤

- 将卷附加到实例

4.7. 在后端之间移动卷

将卷从一个存储后端移动到另一个存储后端有很多原因，例如：

- 停用不再支持的存储系统。
- 更改卷的存储类或层。
- 更改卷的可用区。

使用 Block Storage 服务(cinder)，您可以使用以下方法在后端间移动卷：

- Retype ：只有卷所有者，管理员可以重新键入卷。retype 操作是在后端之间移动卷的最常见方法。如需更多信息，请参阅 块存储卷调整。
- migrate ：只有管理员才能迁移卷。为特定的用例保留卷迁移，因为它受到限制，需要明确了解部署如何工作。如需更多信息，请参阅使用 仪表板在后端间迁移卷， 或使用 CLI 在后端间迁移卷。

限制

红帽支持在可用区内和跨可用区(AZ)间移动卷，但有以下限制：

- 卷必须具有 available 或 in-use 状态才能移动。
- 对使用中的卷的支持取决于驱动程序。
- 卷不能有快照。
- 卷不能属于组或一致性组。

4.7.1. 移动可用卷

您可以在所有后端之间移动可用卷，但性能取决于您使用的后端。许多后端支持协助的迁移。有关协助迁移的后端支持的更多信息，请联系供应商。

支持的迁移可用于卷重新类型和卷迁移。通过协助的迁移，后端可以优化将数据从源后端迁移到目标后端，但两个后端都必须来自同一供应商。

注意

红帽只支持带有多池后端的后端辅助迁移，或者在将 cinder migrate 操作用于单池后端时，如 RBD。

当无法支持在后端之间进行迁移时，块存储服务会执行通用卷迁移。

通用卷迁移要求在 Block Storage (cinder)服务将数据从源卷移动到 Controller 节点前连接两个后端的卷，并从 Controller 节点移到目标卷。无论源和目标后端中的存储类型如何，块存储服务都会无缝执行这个过程。

重要

在执行通用卷迁移前，请确保您有足够的带宽。通用卷迁移的持续时间与卷的大小直接成比例，从而使操作比支持的迁移要慢。

4.7.2. 移动使用的卷

没有可用于移动使用的卷的优化或辅助选项。当您移动使用的卷时，计算服务(nova)必须使用虚拟机监控程序将数据从源后端中的卷传输到目标后端的卷。这需要与运行卷正在使用的实例的虚拟机监控程序协调。

块存储服务(cinder)和计算服务协同工作，以执行此操作。Compute 服务管理大多数工作，因为数据通过 Compute 节点从一个卷复制到另一个卷。

重要

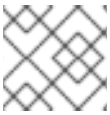
在移动使用卷前，请确保您有足够的带宽。此操作的持续时间与卷的大小直接成比例，从而使操作比支持的迁移要慢。

限制

- 当卷附加到多个 nova 实例时，它们不会被移动。
- 不支持非块设备，将目标后端上的存储协议限制为 iSCSI、光纤通道(FC)和 RBD。

4.8. 块存储卷重新调整

当您重新输入卷时，您可以将卷类型及其设置应用到现有卷。有关卷类型的更多信息，请参阅使用 卷类型分组卷配置。



注意

只有卷所有者和管理员可以重新键入卷。

您可以重新键入新卷类型的额外规格可以应用到现有卷的卷。您可以重新键入卷，以将预定义的设置或存储属性应用到现有卷，例如：

- 将卷移动到不同的后端。
- 更改卷的存储类或层。
- 启用或禁用复制等功能。

卷重新调整是将卷从一个后端移动到另一个后端的标准方法。但是，重新制作卷不一定意味着您必须将卷从一个后端移到另一个后端。然而，在有些情况下，您必须移动卷才能完成重新输入：

- 新卷类型定义了不同的 **volume_backend_name**。
- 当前卷类型的 **volume_backend_name** 未定义，该卷存储在新卷类型的 **volume_backend_name** 指定的后端上。

将卷从一个后端移动到另一个后端可能需要大量时间和资源。因此，当重新类型需要移动数据时，块存储服务默认不会移动数据。除非将迁移策略指定为 retype 请求的一部分，否则操作会失败。如需更多信息，请参阅 Retyping a volume from the Dashboard 或 Retyping a volume from the CLI。

限制

- 您不能重新输入所有卷。有关在后端之间移动卷的更多信息，请参阅 在后端间移动卷。
- 未加密的卷无法重新输入到加密的卷类型，但可将加密的卷重新输入到未加密的卷。
- 不支持将未加密的卷重新设置为相同大小的加密卷，因为加密卷需要额外的空间来存储加密数据。有关加密未加密的卷的更多信息，请参阅 加密未加密的卷。
- 没有管理特权的用户只能重新键入他们拥有的卷。

其他资源

- 创建和配置卷类型

4.8.1. 从仪表板重新定义卷

重新键入卷，以将卷类型及其设置应用到现有卷。

重要

不支持将未加密的卷重新设置为相同大小的加密卷，因为加密卷需要额外的空间来存储加密数据。有关加密未加密的卷的更多信息，请参阅 加密未加密的卷。

先决条件

- 只有卷所有者和管理员可以重新键入卷。
- 访问 Red Hat OpenStack Platform (RHOSP) Dashboard (horizon)。如需更多信息，请参阅 *使用 OpenStack 控制面板管理云资源*。

流程

1. 以 admin 用户或卷所有者登录控制面板。
2. 选择 **Project > Compute > Volumes**。
3. 在您要迁移的卷的 **Actions** 列中，选择 **Change Volume Type**。
4. 在 **Change Volume Type** 对话框中，选择目标卷类型并从 **Type** 列表中定义新后端。
5. 如果要将卷迁移到另一个后端，请从 **Migration Policy** 列表中选择 **On Demand**。如需更多信息，请参阅在后端间移动卷。
6. 点 **Change Volume Type** 开始迁移。

4.8.2. 通过 CLI 重新调整卷

重新键入卷，以将卷类型及其设置应用到现有卷。

重要

不支持将未加密的卷重新设置为相同大小的加密卷，因为加密卷需要额外的空间来存储加密数据。有关加密未加密的卷的更多信息，请参阅 加密未加密的卷。

先决条件

- 只有卷所有者和管理员可以重新键入卷。

流程

1. 提供您的凭据文件。
2. 输入以下命令重新输入卷：

```
$ cinder retype <volume name or id> <new volume type name>
```

3. 如果重新类型操作需要将卷从一个后端移到另一个后端，则块存储服务需要特定的标记：

```
$ cinder retype --migration-policy on-demand <volume name or id> <new volume type name>
```

4.9. 使用控制面板在后端之间迁移卷

使用 Block Storage 服务(cinder)，您可以在可用区和跨可用区(AZ)之间迁移卷。这是将卷从一个后端移到另一个后端的最常用方法。

在高度自定义部署或必须停用存储系统的情况下，管理员可以迁移卷。在这两种用例中，多个存储系统共享相同的 **volume_backend_name**，或者未定义。

限制

- 无法复制卷。
- 目标后端必须与卷的当前后端不同。
- 现有卷类型必须对新后端有效，这意味着必须满足以下条件：
  - 卷类型不能在其额外 specs 中定义 **backend_volume_name**，或者两个块存储后端都必须使用相同的 **backend_volume_name** 配置。
  - 两个后端都必须支持卷类型中配置的相同功能，如支持精简配置、密集置备或其他功能配置。

注意

将卷从一个后端移动到另一个后端可能需要大量时间和资源。如需更多信息，请参阅在后端间移动卷。

先决条件

- 您必须是一个项目管理员才能迁移卷。
- 访问 Red Hat OpenStack Platform (RHOSP) Dashboard (horizon)。如需更多信息，请参阅 *使用 OpenStack 控制面板管理云资源*。

流程

1. 以 admin 用户身份登录控制面板。

2. 选择 **Admin > Volumes**。
3. 在您要迁移的卷的 **Actions** 列中，选择 **Migrate Volume**。
4. 在 **Migrate Volume**对话框中，从 **Destination Host** 下拉列表中选择目标主机。

注意

要绕过主机迁移的任何驱动程序优化，请选择 **Force Host Copy** 复选框。

5. 点 **Migrate** 开始迁移。

4.10. 使用 CLI 在后端之间迁移卷

使用 Block Storage 服务(cinder)，您可以在可用区和跨可用区(AZ)之间迁移卷。这是将卷从一个后端移到另一个后端的最常用方法。

在高度自定义部署或必须停用存储系统的情况下，管理员可以迁移卷。在这两种用例中，多个存储系统共享相同的 **volume_backend_name**，或者未定义。

限制

- 无法复制卷。
- 目标后端必须与卷的当前后端不同。
- 现有卷类型必须对新后端有效，这意味着必须满足以下条件：
  - 卷类型不能在其额外 specs 中定义 **backend_volume_name**，或者两个块存储后端都必须使用相同的 **backend_volume_name** 配置。
  - 两个后端都必须支持卷类型中配置的相同功能，如支持精简配置、密集置备或其他功能配置。

注意

将卷从一个后端移动到另一个后端可能需要大量时间和资源。如需更多信息，请参阅在后端间移动卷。

先决条件

- 您必须是一个项目管理员才能迁移卷。

流程

1. 提供 overcloud 凭证文件：

```
$ source ~/<credentials_file>
```

   - 将 **<credentials_file** > 替换为您的凭据文件的名称，如 **overcloudrc**。

2. 输入以下命令来检索目标后端的名称：

```
$ cinder get-pools --detail
```

```
Property                    | Value

...

| name                      | localdomain@lvmdriver-1#lvmdriver-1
| pool_name                 | lvmdriver-1

...

| volume_backend_name       | lvmdriver-1

...

Property                    | Value

...
                                          |
| name                      | localdomain@lvmdriver-2#lvmdriver-1
| pool_name                 | lvmdriver-1

...

| volume_backend_name       | lvmdriver-1

...
```

目标后端名称使用以下语法： **host@volume_backend_name#pool**。

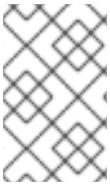
在示例输出中，块存储服务中有两个 LVM 后端：**localdomain@lvmdriver-1#lvmdriver-1** 和 **localdomain@lvmdriver-2#lvmdriver-1**。请注意，两个后端都共享相同的 **volume_backend_name**,**lvmdriver-1**。

3. 输入以下命令将卷从一个后端迁移到另一个后端：

```
$ cinder migrate <volume id or name> <new host>
```

4.11. 管理并取消管理卷及其快照

您可以使用 **cinder manage** 和 **cinder unmanage** 命令将卷添加到块存储卷服务(**cinder-volume**)中或从卷中删除。通常，块存储卷服务管理它所创建的卷，以便它可以，例如，例如，列出、连接和删除这些卷。但是，您可以使用 **cinder unmanage** 命令从块存储服务中删除卷，使其不再列出、附加或删除这个卷。同样，您可以使用 **cinder manage** 命令将卷添加到块存储卷服务，以便它能够用于实例、列出、附加和删除此卷。



注意

如果卷有快照，则无法取消管理。在这种情况下，您必须使用 **cinder snapshot-unmanage** 命令在取消管理卷前取消管理所有快照。同样，当您管理具有快照的卷时，您必须首先管理卷，然后使用 **cinder snapshot-manage** 命令管理快照。

您可以在并行升级 Red Hat OpenStack Platform (RHOSP)部署时使用这些块存储命令，在部署新版本的 RHOSP 时保持现有 RHOSP 版本运行。在这种情况下，您必须取消管理所有快照，然后取消管理卷从现有 RHOSP 中删除卷，然后您必须管理这个卷及其所有快照，将这个卷及其快照添加到新版本的 RHOSP

中。这样，您可以在运行现有云时将卷及其 snaphots 移到新的 RHOSP 版本。

另一种可能的场景是，如果您的一个存储阵列中有一个卷使用裸机，则可能的情况。然后，您决定将在这个机器上运行的软件移到云中，但您仍想使用这个卷。在这种情况下，您可以使用 **cinder manage** 命令将这个卷添加到块存储服务中。

您可以使用 **cinder manageable-list** 命令来确定块存储卷服务的存储阵列中是否有卷没有管理。此列表中的卷通常是用户已经管理的卷，或者已在存储阵列上手动创建的卷，而无需使用块存储卷服务。同样，**cinder snapshot-manageable-list** 命令会列出所有可管理的快照。

cinder manage 命令的语法特定于后端，因为识别卷所需的属性是特定于后端。大多数后端支持 **source-name** 和 **source-id** 属性，其他后端都需要设置额外的属性。有些后端可以列出可管理的卷，以及需要传递哪些参数。有关这些后端，请参阅供应商文档。**cinder unmanage** 命令的语法不特定于后端，您必须指定所需的卷名称或卷 ID。

同样，**cinder snapshot-manage** 命令的语法是特定于后端，因为识别快照所需的属性是特定于后端。**cinder snapshot-unmanage** 命令的语法不特定于后端，您必须指定所需的快照名称或快照 ID。

4.12. 加密未加密的卷

您可以加密未加密的卷。

如果 **cinder-backup** 服务可用，请备份未加密的卷，然后将其恢复到新的加密卷。

如果 **cinder-backup** 服务不可用，则从未加密的卷创建一个 glance 镜像，并从此镜像创建新的加密卷。

先决条件

- 您必须是项目管理员才能创建加密卷。
- 要加密的未加密的卷。

流程

cinder-backup 服务可用：

1. 提供 overcloud 凭证文件：

```
$ source ~/<credentials_file>
```

   - 将 **<credentials_file** > 替换为您的凭据文件的名称，如 **overcloudrc**。

2. 备份当前的未加密的卷：

```
cinder backup-create <unencrypted_volume>
```

   - 将 **<unencrypted_volume** > 替换为未加密的卷的名称或 ID。

3. 创建新加密卷：

```
cinder create <encrypted_volume_size> --volume-type <encrypted_volume_type>
```

   - 将 **<encrypted_volume_size** > 替换为新卷的大小（以 GB 为单位）。这个值必须于未加密卷的大 1 GB，以便容纳加密元数据。
   - 将 **<encrypted_volume_type** > 替换为您需要的加密类型。

4. 将未加密的卷的备份恢复到新的加密卷：

```
cinder backup-restore <backup> --volume <encrypted_volume>
```

- 将 **<backup** > 替换为未加密的卷备份的名称或 ID。
- 将 **<encrypted_volume** > 替换为新加密卷的 ID。

cinder-backup 服务不可用：

1. 提供 overcloud 凭证文件：

```
$ source ~/<credentials_file>
```

- 将 **<credentials_file** > 替换为您的凭据文件的名称，如 **overcloudrc**。

2. 创建未加密的卷的 glance 镜像：

```
cinder upload-to-image <unencrypted_volume> <new_image>
```

- 将 **<unencrypted_volume** > 替换为未加密的卷的名称或 ID。
- 将 **<new_image** > 替换为新镜像的名称。

3. 从比镜像大 1 GB 的镜像创建一个新卷：

```
cinder volume create --size <size> --volume-type luks --image <new_image>
<encrypted_volume_name>
```

- 将 **<** size> 替换为新卷的大小。这个值必须比旧的未加密的卷大 1GB。
- 将 **<new_image** > 替换为您从未加密的卷创建的镜像的名称。
- 将 **<encrypted_volume_name** > 替换为新加密卷的名称。

4.13. RED HAT CEPH STORAGE 后端中的保护和未保护的快照

当您使用 Red Hat Ceph Storage(RHCS)作为 Red Hat OpenStack Platform(RHOSP)部署的后端时，您可以在后端中将快照设置为 **protected**。不要通过 RHOSP 仪表板或 **cinder snapshot-delete** 命令删除受保护的快照，因为删除失败。

当发生这种情况时，首先在 RHCS 后端中将快照设置为 **unprotected**。然后，您可以正常通过 RHOSP 删除快照。

有关保护快照的更多信息，请参阅 Red Hat Ceph Storage *块设备指南*中的保护块设备快照和取消保护块设备快照。

第 5 章 配置 OBJECT STORAGE 服务 (SWIFT)

Red Hat OpenStack Platform (RHOSP) Object Storage 服务(swift)将其对象或数据存储在容器中。容器与文件系统中的目录类似，但它们无法嵌套。容器为用户提供了一种简单的方法来存储任何非结构化数据。例如，对象可以包含 photos、文本文件或镜像。存储的对象不会被压缩。

5.1. 对象存储环

对象存储服务(swift)使用名为 ring 的数据结构来在集群中分发分区空间。这个分区空间是对象存储服务中的数据持久性引擎的核心。通过环，对象存储服务可以快速轻松地在集群中同步每个分区。

Ring 包含有关对象存储分区的信息，以及如何在不同的节点和磁盘间分布分区。当任何对象存储组件与数据交互时，会在环中本地执行快速查找来确定每个对象的可能的分区。

对象存储服务有三个环来存储以下类型的数据：

- 帐户信息
- 容器，以便于将对象组织到帐户下
- 对象副本

5.1.1. 检查集群健康状况

对象存储服务(swift)在后台运行多个进程，以确保长期数据可用性、持久性和持久性。例如：

- 审核员会持续重新读取数据库和对象文件，并使用 checksums 进行比较，以确保没有静默的位。任何不再匹配其校验和的数据库或对象文件都是 quarantined，且在该节点上不可读取。然后，replicators 复制另一个副本，以便再次提供本地副本。
- 当您替换磁盘或节点或对象是 quarantined 时，对象和文件可能会消失。当发生这种情况时，replicators 会将缺少的对象或数据库文件复制到其他节点之一。

对象存储服务包含一个名为 **swift-recon** 的工具，用于从所有节点收集数据并检查整个集群健康状况。

流程

1. 登录到其中一个 Controller 节点。
2. 运行以下命令：

```
[tripleo-admin@overcloud-controller-2 ~]$ sudo podman exec -it -u swift swift_object_server
/usr/bin/swift-recon -arqlT --md5

================================================================================-
-> Starting reconnaissance on 3 hosts (object)
================================================================================
[2018-12-14 14:55:47] Checking async pendings
[async_pending] - No hosts returned valid data.
================================================================================
[2018-12-14 14:55:47] Checking on replication
[replication_failure] low: 0, high: 0, avg: 0.0, total: 0, Failed: 0.0%, no_result: 0, reported: 3
[replication_success] low: 0, high: 0, avg: 0.0, total: 0, Failed: 0.0%, no_result: 0, reported: 3
[replication_time] low: 0, high: 0, avg: 0.0, total: 0, Failed: 0.0%, no_result: 0, reported: 3
[replication_attempted] low: 0, high: 0, avg: 0.0, total: 0, Failed: 0.0%, no_result: 0, reported:
```

```
3
Oldest completion was 2018-12-14 14:55:39 (7 seconds ago) by 198.51.100.186:6000.
Most recent completion was 2018-12-14 14:55:42 (4 seconds ago) by 198.51.100.174:6000.
========================================================================
[2018-12-14 14:55:47] Checking load averages
[5m_load_avg] low: 1, high: 2, avg: 2.1, total: 6, Failed: 0.0%, no_result: 0, reported: 3
[15m_load_avg] low: 2, high: 2, avg: 2.6, total: 7, Failed: 0.0%, no_result: 0, reported: 3
[1m_load_avg] low: 0, high: 0, avg: 0.8, total: 2, Failed: 0.0%, no_result: 0, reported: 3
========================================================================
[2018-12-14 14:55:47] Checking ring md5sums
3/3 hosts matched, 0 error[s] while checking hosts.
========================================================================
[2018-12-14 14:55:47] Checking swift.conf md5sum
3/3 hosts matched, 0 error[s] while checking hosts.
========================================================================
[2018-12-14 14:55:47] Checking quarantine
[quarantined_objects] low: 0, high: 0, avg: 0.0, total: 0, Failed: 0.0%, no_result: 0, reported: 3
[quarantined_accounts] low: 0, high: 0, avg: 0.0, total: 0, Failed: 0.0%, no_result: 0, reported:
3
[quarantined_containers] low: 0, high: 0, avg: 0.0, total: 0, Failed: 0.0%, no_result: 0,
reported: 3
========================================================================
[2018-12-14 14:55:47] Checking time-sync
3/3 hosts matched, 0 error[s] while checking hosts.
========================================================================
```

3. 可选： 使用 **--all** 选项返回附加输出。
   此命令查询 ring 上的所有服务器以获取以下数据：

   - async pendings ：如果集群负载过高，且进程无法快速更新数据库文件，则一些更新会异步进行。这些数字会随时间而减少。
   - 复制指标 ：查看复制时间戳；完整复制通过频繁出现几个错误。旧条目（例如，带有 6 个月前的时间戳的条目）表示节点上的复制还没有在最后的六个月内完成。
   - Ring md5sums：这样可确保所有环文件在所有节点上都一致。
   - **swift.conf** md5sums：这样可确保所有配置文件在所有节点上都一致。
   - Quarantined 文件：所有节点必须具有 no 或非常少的 quarantined 文件。
   - time-sync ：所有节点必须同步。

5.1.2. 增加环分区电源

Ring power 决定将资源（如帐户、容器或对象）映射到的分区。分区包含在资源存储在后端文件系统中的路径中。因此，更改分区电源需要将资源重新定位到后端文件系统中的新路径。

在大量填充的集群中，重新定位过程会非常耗时。为避免停机，在集群仍在运行时重新定位资源。您必须在不临时丢失数据访问或损害进程性能（如复制和审核）的情况下执行此操作。如需增加环分区电源的帮助，请联系红帽支持。

5.1.3. 对象存储服务的分区电源建议

使用单独的 Red Hat OpenStack Platform (RHOSP) Object Storage 服务(swift)节点时，使用较高的分区电源值。

对象存储服务使用修改后的 *哈希环* 在磁盘和节点之间分发数据。默认有三个环：一个用于帐户，一个用于容器，另一个用于对象。每个环使用一个称为 *分区电源* 的固定参数。此参数设置可创建的最大分区数。

分区电源参数非常重要，只能为新容器及其对象更改。因此，*在初始部署* 前设置这个值非常重要。

对于 RHOSP director 部署的环境，默认分区 power 值为 **10**。对于较小的部署来说，这是合理的值，特别是您只计划在 Controller 节点上为对象存储服务使用磁盘。

如果使用三个副本，下表可帮助您选择适当的分区电源：

表 5.1. 每个可用磁盘数量的适当分区电源值

分区 Power	磁盘的最大数量
10	~ 35
11	~ 75
12	~ 150
13	~ 250
14	~ 500

重要

设置过度高的分区电源值（例如，**14** 代表 40 个磁盘）对复制时间造成负面影响。

要设置分区电源，请使用以下资源：

```
parameter_defaults:
  SwiftPartPower: 11
```

提示

您还可以为新容器配置额外的对象服务器 ring。如果要在最初使用低分区电源的对象存储服务部署中添加更多磁盘，这非常有用。

其他资源

- swift 上游文档中的 Rings
- *使用 director 安装和管理 Red Hat OpenStack Platform* 中的 修改 overcloud 环境

5.1.4. 自定义环

随着技术进步和存储容量的增加，创建自定义环是更新现有 Object Storage 集群的方法。

当您向集群添加新节点时，它们的特征可能与原始节点的不同。如果没有自定义调整，可能会使用新节点的较大的容量。或者，如果对环中的权重发生变化，数据分布可能会变得不均匀，这会降低安全性。

自动化可能无法与未来的技术趋势保持同步。例如，现在使用的一些较旧的 Object Storage 集群以前源自在 SSD 可用之前。

在集群不断增长和技术时，ring 构建器有助于管理对象存储。如需创建自定义环的帮助，请联系红帽支持。

5.2. 自定义对象存储服务

根据 Red Hat OpenStack Platform (RHOSP)环境的要求，您可能需要自定义 Object Storage 服务(swift)的一些默认设置来优化部署性能。

5.2.1. 配置 fast-post

默认情况下，Object Storage 服务(swift)会在其元数据的任何部分更改时完整复制对象。您可以使用 *fast-post* 功能来防止这种情况。当更改多个大型对象的内容类型时，fast-post 功能会节省时间。

要启用 fast-post 功能，请在 Object Storage 代理服务中禁用 **object_post_as_copy** 选项。

流程

1. 编辑 **swift_params.yaml** ：

```
$ cat > swift_params.yaml << EOF
  parameter_defaults:
      ExtraConfig:
        swift::proxy::copy::object_post_as_copy: False
  EOF
```

2. 部署或更新 overcloud 时包含参数文件：

```
$ openstack overcloud deploy [... previous args ...] \
  -e swift_params.yaml
```

5.2.2. 启用 at-rest 加密

默认情况下，上传到 Object Storage 服务(swift)的对象是未加密的。因此，可以从文件系统中直接访问对象。如果在磁盘被丢弃前没有正确清除磁盘，这可能会带来安全风险。对象存储对象。如需更多信息，请参阅 *使用 Key Manager 服务管理 secret* 中的加密 Object Storage (swift) at-rest 对象 。

5.2.3. 部署独立对象存储服务集群

您可以使用可组合角色概念来部署独立 Object Storage 服务(swift)集群，其最小其他服务，如 OpenStack Identity service (keystone)或 HAProxy。

流程

1. 从 **/usr/share/openstack-tripleo-heat-templates** 复制 **roles_data.yaml**。
2. 编辑新文件。
3. 删除不需要的控制器角色，如 Aodh*、Ceilometer、Ceph*、Cinder*、Glance*、Ironic*、Manila*、Nova*、Octaana、Octaana、Swift*。

4. 在 **roles_data.yaml** 中查找 ObjectStorage 角色。
5. 将此角色复制到同一文件中的新角色，并将其命名为 **ObjectProxy**。
6. 将 **SwiftStorage** 替换为此角色中的 **SwiftProxy**。
   以下 **roles_data.yaml** 文件示例显示了示例角色。

```
- name: Controller
  description: |
 Controller role that has all the controller services loaded and handles
 Database, Messaging and Network functions.
  CountDefault: 1
  tags:
 - primary
 - controller
  networks:
 - External
 - InternalApi
 - Storage
 - StorageMgmt
 - Tenant
  HostnameFormatDefault: '%stackname%-controller-%index%'
  ServicesDefault:
 - OS::TripleO::Services::AuditD
 - OS::TripleO::Services::CACerts
 - OS::TripleO::Services::CertmongerUser
 - OS::TripleO::Services::Clustercheck
 - OS::TripleO::Services::Docker
 - OS::TripleO::Services::Ec2Api
 - OS::TripleO::Services::Etcd
 - OS::TripleO::Services::HAproxy
 - OS::TripleO::Services::Keepalived
 - OS::TripleO::Services::Kernel
 - OS::TripleO::Services::Keystone
 - OS::TripleO::Services::Memcached
 - OS::TripleO::Services::MySQL
 - OS::TripleO::Services::MySQLClient
 - OS::TripleO::Services::Ntp
 - OS::TripleO::Services::Pacemaker
 - OS::TripleO::Services::RabbitMQ
 - OS::TripleO::Services::Securetty
 - OS::TripleO::Services::Snmp
 - OS::TripleO::Services::Sshd
 - OS::TripleO::Services::Timezone
 - OS::TripleO::Services::TripleoFirewall
 - OS::TripleO::Services::TripleoPackages
 - OS::TripleO::Services::Vpp

- name: ObjectStorage
  CountDefault: 1
  description: |
 Swift Object Storage node role
  networks:
 - InternalApi
 - Storage
 - StorageMgmt
```

```
  disable_upgrade_deployment: True
  ServicesDefault:
 - OS::TripleO::Services::AuditD
 - OS::TripleO::Services::CACerts
 - OS::TripleO::Services::CertmongerUser
 - OS::TripleO::Services::Collectd
 - OS::TripleO::Services::Docker
 - OS::TripleO::Services::FluentdClient
 - OS::TripleO::Services::Kernel
 - OS::TripleO::Services::MySQLClient
 - OS::TripleO::Services::Ntp
 - OS::TripleO::Services::Securetty
 - OS::TripleO::Services::SensuClient
 - OS::TripleO::Services::Snmp
 - OS::TripleO::Services::Sshd
 - OS::TripleO::Services::SwiftRingBuilder
 - OS::TripleO::Services::SwiftStorage
 - OS::TripleO::Services::Timezone
 - OS::TripleO::Services::TripleoFirewall
 - OS::TripleO::Services::TripleoPackages

- name: ObjectProxy
  CountDefault: 1
  description: |
 Swift Object proxy node role
  networks:
 - InternalApi
 - Storage
 - StorageMgmt
  disable_upgrade_deployment: True
  ServicesDefault:
 - OS::TripleO::Services::AuditD
 - OS::TripleO::Services::CACerts
 - OS::TripleO::Services::CertmongerUser
 - OS::TripleO::Services::Collectd
 - OS::TripleO::Services::Docker
 - OS::TripleO::Services::FluentdClient
 - OS::TripleO::Services::Kernel
 - OS::TripleO::Services::MySQLClient
 - OS::TripleO::Services::Ntp
 - OS::TripleO::Services::Securetty
 - OS::TripleO::Services::SensuClient
 - OS::TripleO::Services::Snmp
 - OS::TripleO::Services::Sshd
 - OS::TripleO::Services::SwiftRingBuilder
 - OS::TripleO::Services::SwiftProxy
 - OS::TripleO::Services::Timezone
 - OS::TripleO::Services::TripleoFirewall
 - OS::TripleO::Services::TripleoPackages
```

7. 使用常规的 **openstack deploy** 命令部署 overcloud，包括新角色。

```
$ openstack overcloud deploy --templates -r roles_data.yaml -e [...]
```

5.2.4. 使用外部 SAN 磁盘

默认情况下，当 Red Hat OpenStack Platform (RHOSP) director 部署 Object Storage 服务(swift)时，对象存储会被配置和优化以使用独立本地磁盘。此配置可确保工作负载在所有磁盘上分布，这有助于在节点故障或其他系统问题期间最大程度降低性能影响。

在性能影响事件时，使用单个 SAN 的环境可能会遇到所有 LUN 的性能降低。对象存储服务无法缓解使用 SAN 磁盘的环境中的性能问题。因此，为对象存储使用额外的本地磁盘来满足性能和磁盘空间要求。如需更多信息，请参阅 Object Storage 服务的磁盘建议 。

将外部 SAN 用于对象存储需要根据具体情况进行评估。如需更多信息，请联系红帽支持团队。

重要

如果您选择将外部 SAN 用于对象存储，请注意以下条件：

- 红帽不为与使用外部 SAN 进行对象存储造成的性能相关的问题提供支持。
- 红帽不提供对核心对象存储服务产品之外的问题提供支持。要获得高可用性和性能的支持，请联系您的存储厂商。
- 红帽没有使用对象存储服务测试 SAN 解决方案。有关第三方产品的兼容性、指导和支持的更多信息，请联系您的存储供应商。
- 红帽建议您使用部署评估和测试性能需求。要确认您的 SAN 部署经过测试、受支持并满足您的性能要求，请联系您的存储供应商。

流程

- 此模板是如何将两个设备(**/dev/mapper/vdb** 和 **/dev/mapper/vdc**)用于对象存储的示例：

```
parameter_defaults:
  SwiftMountCheck: true
  SwiftUseLocalDir: false
  SwiftRawDisks: {"vdb": {"base_dir":"/dev/mapper/"}, \
                  "vdc": {"base_dir":"/dev/mapper/"}}
```

5.2.5. 对象存储服务的磁盘建议

为 Red Hat OpenStack Platform (RHOSP) Object Storage 服务(swift)使用一个或多个单独的本地磁盘。

默认情况下，RHOSP director 将系统磁盘上的 **/srv/node/d1** 目录用于对象存储服务。在 Controller 上，其他服务也会使用这个磁盘，磁盘可能会成为性能瓶颈。

以下示例是 RHOSP 编排服务(heat)自定义环境文件摘录。在每个 Controller 节点上，对象存储服务使用两个独立磁盘。两个磁盘的完整都包含一个 XFS 文件系统：

```
parameter_defaults:
  SwiftRawDisks: {"sdb": {}, "sdc": {}}
```

SwiftRawDisks 定义节点上的每个存储磁盘。这个示例在每个 Controller 节点上同时定义了 **sdb** 和 **sdc** 磁盘。

重要

在配置多个磁盘时，请确保 Bare Metal 服务(ironic)使用预期的根磁盘。

其他资源

- *使用 director 安装和管理 Red Hat OpenStack Platform* 指南中的 为多磁盘集群定义根磁盘。

5.3. 添加或删除对象存储节点

要向集群添加新的 Object Storage (swift)节点，您必须增加节点数，更新环，并同步更改。您可以通过添加节点到 overcloud 或扩展裸机节点来增加节点数。

要从集群中删除 Object Storage 节点，您可以执行简单删除或增量删除，具体取决于集群中的数据量。

5.3.1. 向 overcloud 添加节点

您可以将更多节点添加到 overcloud。

注意

全新安装 Red Hat OpenStack Platform (RHOSP)不包括某些更新，如安全勘误和程序错误修复。因此，如果您要扩展使用红帽客户门户网站或 Red Hat Satellite Server 的连接环境，RPM 更新不会应用到新节点。要将最新的更新应用到 overcloud 节点，您必须执行以下操作之一：

- 在横向扩展操作后，完成节点的 overcloud 更新。
- 在 scale-out 操作前，使用 **virt-customize** 工具将软件包修改为基础 overcloud 镜像。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。

流程

1. 创建名为 **newnodes.json** 的新 JSON 文件，其中包含您要注册的新节点的详情：

```
{
  "nodes":[
    {
        "mac":[
            "dd:dd:dd:dd:dd:dd"
        ],
        "cpu":"4",
        "memory":"6144",
        "disk":"40",
        "arch":"x86_64",
        "pm_type":"ipmi",
        "pm_user":"admin",
        "pm_password":"p@55w0rd!",
        "pm_addr":"192.02.24.207"
    },
    {
        "mac":[
            "ee:ee:ee:ee:ee:ee"
        ],
        "cpu":"4",
        "memory":"6144",
        "disk":"40",
        "arch":"x86_64",
```

```
        "pm_type":"ipmi",
        "pm_user":"admin",
        "pm_password":"p@55w0rd!",
        "pm_addr":"192.02.24.208"
    }
  ]
}
```

2. 以 **stack** 用户身份登录 undercloud 主机。
3. 查找 **stackrc** undercloud 凭证文件：

```
$ source ~/stackrc
```

4. 注册新节点：

```
$ openstack overcloud node import newnodes.json
```

5. 为每个新节点启动内省过程：

```
$ openstack overcloud node introspect \
 --provide <node_1> [<node_2>] [<node_n>]
```

- 使用 **--provide** 选项，在内省后将所有指定的节点重置为 **available** 状态。
- 将 **<node_1&** gt **; , <node_** 2>，将直到 < **node_n** > 的所有节点替换为您要内省的每个节点的 UUID。

6. 为每个新节点配置镜像属性：

```
$ openstack overcloud node configure <node>
```

5.3.2. 扩展裸机节点

要增加现有 overcloud 中的裸机节点数量，请在 **overcloud-baremetal-deploy.yaml** 文件中增加节点数并重新部署 overcloud。

先决条件

- 新的裸机节点已注册、内省，并可用于调配和部署。如需更多信息，请参阅为 overcloud 注册节点 和 创建裸机节点硬件清单。

流程

1. 以 **stack** 用户身份登录 undercloud 主机。
2. 查找 **stackrc** undercloud 凭证文件：

```
$ source ~/stackrc
```

3. 打开用于置备裸机节点的 **overcloud-baremetal-deploy.yaml** 节点定义文件。
4. 增加您要扩展的角色的 **count** 参数。例如，以下配置将 Object Storage 节点数增加到 4：

```
-
```

```
- name: Controller
  count: 3
- name: Compute
  count: 10
- name: ObjectStorage
  count: 4
```

5. 可选：为新节点配置预测节点放置。例如，使用以下配置在 **node03** 上置备新的 Object Storage 节点：

```
- name: ObjectStorage
  count: 4
  instances:
  - hostname: overcloud-objectstorage-0
    name: node00
  - hostname: overcloud-objectstorage-1
    name: node01
  - hostname: overcloud-objectstorage-2
    name: node02
  - hostname: overcloud-objectstorage-3
    name: node03
```

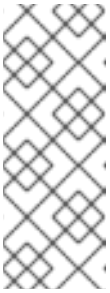
6. 可选：定义您要分配给新节点的任何其他属性。有关您可以在节点定义文件中配置节点属性的属性的更多信息，请参阅 裸机节点置备属性。

7. 如果您使用 Object Storage 服务(swift)和整个磁盘 overcloud 镜像，**overcloud-hardened-uefi-full**，请根据您的磁盘大小配置 **/srv** 分区的大小以及 **/var** 和 **/srv** 的存储要求。如需更多信息，请参阅为对象存储服务配置整个磁盘分区。

8. 置备 overcloud 节点：

```
$ openstack overcloud node provision \
  --stack <stack> \
  --network-config \
  --output <deployment_file> \
  /home/stack/templates/overcloud-baremetal-deploy.yaml
```

- 将 **&** lt;stack> 替换为置备裸机节点的堆栈名称。如果未指定，则默认为 **overcloud**。
- 包含 **--network-config** 参数，为 **cli-overcloud-node-network-config.yaml** Ansible playbook 提供网络定义。
- 将 **<deployment_file>** 替换为用于部署命令生成的 heat 环境文件的名称，如 **/home/stack/templates/overcloud-baremetal-deployed.yaml**。



注意

如果您从 Red Hat OpenStack Platform 16.2 升级到 17.1，则必须在 **openstack overcloud node provision** 命令中在升级过程中创建或更新的 YAML 文件。例如，使用 **/home/stack/tripleo-[stack]-baremetal-deploy.yaml** 文件，而不是 **/home/stack/templates/overcloud-baremetal-deployed.yaml** 文件。有关更多信息，请参阅 执行 overcloud 的采用和 *准备* Framework (16.2 到 17.1)。

9. 在一个单独的终端中监控置备进度。当置备成功时，节点状态将从 **available** 变为 **active** ：

```
$ watch openstack baremetal node list
```

10. 使用其他环境文件将生成的 **overcloud-baremetal-deployed.yaml** 文件添加到堆栈中，并部署 overcloud：

```
$ openstack overcloud deploy --templates \
  -e [your environment files] \
  -e /home/stack/templates/overcloud-baremetal-deployed.yaml \
  --deployed-server \
  --disable-validations \
  ...
```

5.3.3. 定义专用对象存储节点

为 Red Hat OpenStack Platform (RHOSP)对象存储服务指定额外的节点以提高性能。

如果您要将额外的节点专用于对象存储服务，请编辑自定义 **roles_data.yaml** 文件，从 Controller 节点中删除 Object Storage 服务条目。具体来说，从 **Controller** 角色的 **ServicesDefault** 列表中删除以下行：

```
- OS::TripleO::Services::SwiftStorage
```

5.3.4. 更新并重新平衡对象存储环

对象存储服务(swift)需要所有 Controller 和 Object Storage 节点上的相同环文件。如果替换 Controller 节点或 Object Storage 节点，则必须在 overcloud 更新后同步这些节点或 Object Storage 节点，以确保正常工作。

流程

1. 以 **stack** 用户身份登录 undercloud，再创建一个临时目录：

```
$ mkdir temp && cd temp/
```

2. 将 overcloud 环文件从之前现有节点之一（本例中为 controller 0）下载到新目录中：

```
$ ssh tripleo-admin@overcloud-controller-0.ctlplane 'sudo tar -czvf - \
  /var/lib/config-data/puppet-generated/swift_ringbuilder/etc/swift \
  /{*.builder,*.ring.gz,backups/*.builder}' > swift-rings.tar.gz
```

3. 提取环并更改到 ring 子目录：

```
$ tar xzvf swift-rings.tar.gz && cd \
var/lib/config-data/puppet-generated/swift_ringbuilder/etc/swift/
```

4. 根据您的设备详情收集以下变量的值：

- **<device_name>**:

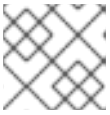
```
$ openstack baremetal introspection data save \
  <node_name> | jq ".inventory.disks"
```

- **<node_ip>**:

```
$ metalsmith <node_name> show
```

- **<port** >: 默认端口为 **600x**。如果您更改了默认值，请使用适用的端口。
- **<builder_file** > ：第 3 步中的构建器文件名。
- **<weight** &gt **; 和 <zone** > 变量是用户定义的。

5. 使用 **swift-ring-builder** 将新节点添加到现有环中。根据设备详情替换变量。



注意

您必须安装 **python3-swift** RPM 以使用 **swift-ring-builder** 命令。

```
$ swift-ring-builder etc/swift/<builder_file> \
  add <zone>-<node_ip>:<port>/<device_name> <weight>
```

6. 重新平衡环，以确保使用新设备：

```
$ swift-ring-builder etc/swift/<builder_file> rebalance
```

7. 将修改后的环文件上传到 Controller 节点，并确保使用这些环文件。使用类似以下示例的脚本来分发环文件：

```
#!/bin/sh
set -xe

ALL="tripleo-admin@overcloud-controller-0.ctlplane \
tripleo-admin@overcloud-controller-1.ctlplane \
tripleo-admin@overcloud-controller-2.ctlplane"
```

- 将环上传到所有节点并重新启动对象存储服务：

```
for DST in ${ALL}; do
  cat swift-rings.tar.gz | ssh "${DST}" 'sudo tar -C / -xvzf -'
  ssh "${DST}" 'sudo podman restart swift_copy_rings'
  ssh "${DST}" 'sudo systemctl restart tripleo_swift*'
done
```

5.3.5. 同步节点更改并迁移数据

在将新环文件复制到其正确的文件夹后，您必须将更改的环文件发送到 Object Storage (swift)容器。

重要的

- 不要同时迁移所有数据。以 10% 的增量迁移数据。例如，将源设备的权重配置为等于 90.0，将目标设备配置为 equal 10.0。然后，将源设备的权重配置为等于 80.0 和 20.0。继续逐步迁移数据，直到您完成此过程。在迁移过程中，如果您同时移动所有数据，旧数据位于源设备上，但环会指向所有副本的新目标设备。在 replicators 将所有数据移到目标设备前，数据将无法访问。

- 在迁移过程中，Object Storage 环会重新分配数据的位置，然后 replicator 将数据移到新位置。当集群活动增加时，因为负载增加，复制过程会减慢。集群越大，复制传递所需的时间越长。这是预期的行为，但如果客户端访问当前被重新定位的数据，则日志文件中可能会导致 404 错误。当代理尝试从新位置检索数据，但数据还没有在新位置上时，**swift-proxy** 会在日志文件中报告 404 错误。
  当迁移逐渐时，代理会访问没有移动的副本，且不会发生错误。当代理尝试从其他副本检索数据时，日志文件中的 404 错误会解决。要确认复制过程正在进行，请参阅复制日志。Object Storage 服务(swift)每五分钟发出复制日志。

流程

1. 使用类似以下示例的脚本，将环文件从之前现有的 Controller 节点分发到所有 Controller 节点，并在这些节点上重启对象存储服务容器：

```
#!/bin/sh
set -xe

SRC="tripleo-admin@overcloud-controller-0.ctlplane"
ALL="tripleo-admin@overcloud-controller-0.ctlplane \
tripleo-admin@overcloud-controller-1.ctlplane \
tripleo-admin@overcloud-controller-2.ctlplane"
```

   - 获取当前的环文件集合：

```
ssh "${SRC}" 'sudo tar -czvf - \
  /var/lib/config-data/puppet-generated/swift_ringbuilder/etc/swift \
  /{*.builder,*.ring.gz,backups/*.builder}' > swift-rings.tar.gz
```

   - 将环上传到所有节点并重新启动对象存储服务：

```
for DST in ${ALL}; do
  cat swift-rings.tar.gz | ssh "${DST}" 'sudo tar -C / -xvzf -'
  ssh "${DST}" 'sudo podman restart swift_copy_rings'
  ssh "${DST}" 'sudo systemctl restart tripleo_swift*'
done
```

2. 要确认数据正在移至新磁盘，请在新存储节点上运行以下命令：

```
$ sudo grep -i replication  /var/log/container/swift/swift.log
```

5.3.6. 删除 Object Storage 节点

删除 Object Storage (swift)节点的方法有两种：

- 简单删除：此方法在一个操作中删除节点，并适用于具有较小的数据量的高效集群。
- 增量删除 ：更改环，以减少您要删除的节点上的磁盘的权重。如果要最大程度降低对存储网络使用量的影响，或者集群包含更大的数据量，则这种方法是适当的。

对于这两种方法，您遵循 **缩减裸机节点** 的步骤。但是，为了进行增量删除，请完成这些先决条件来更改存储环，以减少您要删除的节点上的磁盘权重：

先决条件

- 对象存储环已更新并重新平衡。如需更多信息，请参阅更新和重新平衡对象存储环。
- Object Storage 环中的更改会被同步。如需更多信息，请参阅 同步节点更改和迁移数据。

有关替换对象存储节点的详情，请查看 **缩减裸机节点** 开始时的先决条件。

5.3.7. 缩减裸机节点

若要缩减 overcloud 中的裸机节点数量，请标记您要从节点定义文件的堆栈中删除的节点，重新部署 overcloud，然后从 overcloud 中删除裸机节点。

先决条件

- 成功安装 undercloud。有关更多信息，请参阅在 undercloud 上安装 director。
- 成功部署 overcloud。如需更多信息，请参阅 使用预置备节点配置基本 overcloud。
- 如果要替换 Object Storage 节点，请将您要删除的节点中的数据复制到新替换节点。等待新节点上复制传递完成。在 **/var/log/swift/swift.log** 文件中检查复制传递进度。传递完成后，Object Storage 服务(swift)会将条目添加到类似以下示例的日志中：

```
Mar 29 08:49:05 localhost object-server: Object replication complete.
Mar 29 08:49:11 localhost container-server: Replication run OVER
Mar 29 08:49:13 localhost account-server: Replication run OVER
```

流程

1. 以 **stack** 用户身份登录 undercloud 主机。
2. 查找 **stackrc** undercloud 凭证文件：

```
$ source ~/stackrc
```

3. 对于您要缩减的角色，减少 **overcloud-baremetal-deploy.yaml** 文件中的 **count** 参数。
4. 定义您要从堆栈中删除的每个节点的 **主机名和 名称** （如果它们尚未在角色的 **instances** 属性中定义）。
5. 将 attribute **provisioned: false** 添加到您要删除的节点。例如，要从堆栈中删除节点 **overcloud-objectstorage-1**，请在 **overcloud-baremetal-deploy.yaml** 文件中包含以下代码片段：

```
- name: ObjectStorage
  count: 3
  instances:
  - hostname: overcloud-objectstorage-0
    name: node00
  - hostname: overcloud-objectstorage-1
    name: node01
    # Removed from cluster due to disk failure
    provisioned: false
  - hostname: overcloud-objectstorage-2
    name: node02
  - hostname: overcloud-objectstorage-3
    name: node03
```

重新部署 overcloud 后，堆栈中不再存在使用 **provisioned: false** 属性定义的节点。但是，这些节点仍然以置备状态运行。

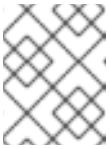
注意

要从堆栈中临时删除节点，请使用 **provisioned: false** 属性部署 overcloud，然后使用 **provisioned: true** 属性重新部署 overcloud，以将节点返回到堆栈。

6. 从 overcloud 删除节点：

```
$ openstack overcloud node delete \
 --stack <stack> \
 --baremetal-deployment \
  /home/stack/templates/overcloud-baremetal-deploy.yaml
```

- 将 **&** lt;stack> 替换为置备裸机节点的堆栈名称。如果未指定，则默认为 **overcloud**。



注意

不要将您要从堆栈中删除的节点作为命令参数包括在 **openstack overcloud node delete** 命令中。

7. 删除 ironic 节点：

```
$ openstack baremetal node delete <IRONIC_NODE_UUID>
```

将 **IRONIC_NODE_UUID** 替换为节点的 UUID。

8. 置备 overcloud 节点以生成更新的 heat 环境文件，以包括在部署命令中：

```
$ openstack overcloud node provision \
 --stack <stack> \
 --output <deployment_file> \
 /home/stack/templates/overcloud-baremetal-deploy.yaml
```

- 将 **<deployment_file>** 替换为用于部署命令生成的 heat 环境文件的名称，如 **/home/stack/templates/overcloud-baremetal-deployed.yaml**。

9. 将 provisioning 命令生成的 **overcloud-baremetal-deployed.yaml** 文件添加到与其他环境文件的堆栈中，并部署 overcloud：

```
$ openstack overcloud deploy \
 ...
 -e /usr/share/openstack-tripleo-heat-templates/environments \
 -e /home/stack/templates/overcloud-baremetal-deployed.yaml \
 --deployed-server \
 --disable-validations \
 ...
```

5.4. 对象存储服务中的容器管理

为帮助满足对象存储服务(swift)中的组织，您可以使用伪文件夹。这些文件夹是可包含对象和嵌套的逻辑设备。例如，您可以创建一个 *Images* 文件夹，在其中存储图片和用于存储视频的 *Media* 文件夹。

您可以在各个项目中创建一个或多个容器，以及每个容器中的一个或多个对象或伪文件夹。

5.4.1. 创建私有和公共容器

使用控制面板在 Object Storage 服务(swift)中创建容器。

流程

1. 在控制面板中，选择 **Project > Object Store > Containers**。
2. 点 **Create Container**。
3. 指定 **Container Name**，然后在 **Container Access** 字段中选择以下之一。

类型	描述
私有	限制当前项目中用户的访问权限。
公开	允许 API 访问具有公共 URL 的任何人。但是，在控制面板中，项目用户无法看到来自其他项目的公共容器和数据。

4. 点 **Create Container**。
5. 可选：新容器使用默认存储策略。如果您定义了多个存储策略，例如，一个默认策略以及启用纠删代码的另一个策略，您可以将容器配置为使用非默认存储策略：

```
$ swift post -H "X-Storage-Policy:<policy>" <container_name>
```

- 将 &**lt;policy** > 替换为您要容器使用的策略的名称或别名。
- 将 **<container_name** > 替换为容器的名称。

5.4.2. 为容器创建伪文件夹

使用控制面板，在 Object Storage 服务(swift)中为容器创建一个伪文件夹。

流程

1. 在控制面板中，选择 **Project > Object Store > Containers**。
2. 单击您要添加伪文件夹的容器的名称。
3. 单击 **Create Pseudo-folder**。
4. 在 **Pseudo-folder Name** 字段中指定名称，然后单击 **Create**。

5.4.3. 从对象存储服务中删除容器

使用控制面板从对象存储服务(swift)中删除容器。

流程

1. 在控制面板中，选择 **Project > Object Store > Containers**。

2. 浏览 **Containers** 部分中的容器，并确保所有对象都已删除。如需更多信息，请参阅从对象存储服务中删除对象。
3. 在容器箭头菜单中选择 **Delete Container**。
4. 点 **Delete Container** 以确认删除容器。

5.4.4. 将对象上传到容器

如果您没有将实际文件上传到对象存储服务(swift)，则对象仍然作为可以稍后用于上传文件的占位符创建。

流程

1. 在控制面板中，选择 **Project > Object Store > Containers**。
2. 单击要放置上传对象的容器的名称。如果容器中已存在伪文件夹，您可以单击其名称。
3. 浏览您的文件，然后单击 **Upload Object**。
4. 在 **Object Name** 字段中指定名称：
   - 您可以使用 **/** 字符（如 *Images/myImage.jpg* ）在名称中指定伪文件夹。如果指定的文件夹不存在，则会在对象上传时创建它。
   - 不是位置唯一的名称（即对象已存在）会覆盖对象的内容。
5. 单击 **Upload Object**。

5.4.5. 在容器间复制对象

使用控制面板在对象存储服务(swift)中复制对象。

流程

1. 在控制面板中，选择 **Project > Object Store > Containers**。
2. 单击对象的容器或文件夹的名称（以显示对象）。
3. 单击 **Upload Object**。
4. 浏览要复制的文件，然后在其箭头菜单中选择 **Copy**。
5. 指定以下内容：

字段	描述
目标容器	新对象的目标容器。
路径	目标容器中的伪文件夹；如果文件夹尚不存在，则会创建它。
目标对象名称	新对象的名称。如果您使用不对位置唯一的名称（即对象已存在），它会覆盖对象之前的内容。

6. 单击 **Copy Object**。

5.4.6. 从对象存储服务中删除对象

使用控制面板从对象存储服务(swift)中删除对象。

流程

1. 在控制面板中，选择 **Project > Object Store > Containers**。
2. 浏览对象，然后在其箭头菜单中选择 **Delete Object**。
3. 单击 **Delete Object** 以确认对象已被删除。

第 6 章 配置共享文件系统服务(MANILA)

使用共享文件系统服务(manila)，您可以置备多个云用户实例、裸机节点或容器可以使用的共享文件系统。云管理员创建共享类型，以准备共享服务，并使最终用户能够创建和管理共享。您可以使用共享文件系统服务命令行客户端来管理共享文件系统。

先决条件

- 最终用户至少需要一种共享类型才能使用共享文件系统服务。
- 对于 **driver_handles_share_servers=false** 的后端，云管理员会提前配置必要的网络，而不是动态地在共享文件系统后端中。
- 对于 CephFS-NFS 后端，云管理员使用隔离网络和环境参数以及自定义 **network_data** 文件部署 Red Hat OpenStack Platform (RHOSP) director，以便为 NFS 导出创建隔离的 StorageNFS 网络。部署后，在 overcloud 使用之前，管理员创建对应的 Networking 服务(neutron) StorageNFS 共享提供商网络，映射到数据中心的隔离 StorageNFS 网络。
- 要使 Compute 实例连接到此共享提供商网络，用户必须添加额外的 neutron 端口。

6.1. 配置共享文件系统服务后端

当云管理员使用 Red Hat OpenStack Platform (RHOSP) director 部署共享文件系统服务(manila)时，他们可以选择一个或多个支持的后端，如原生 CephFS、CephFS-NFS、NetApp、Dell EMC unity 等。

有关原生 CephFS 和 CephFS-NFS 的更多信息，请参阅 *部署 Red Hat Ceph Storage 和 Red Hat OpenStack Platform*。

有关支持的后端设备和驱动程序的完整列表，请参阅 Red Hat OpenStack Platform 中的红帽知识库文章、组件、插件和驱动程序支持的 Manila 部分。

6.1.1. 配置多个后端

后端是与共享文件系统服务(manila)驱动程序配对的存储系统或技术，用于导出文件系统。共享文件系统服务要求至少一个后端才能运行。在很多情况下，一个后端就足够了。但是，您也可以在单个共享文件系统服务安装中使用多个后端。

重要

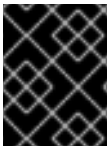
Red Hat OpenStack Platform (RHOSP)不支持到共享文件系统服务部署的同一后端的多个实例。例如，您无法在同一部署中添加两个 Red Hat Ceph Storage 集群作为后端。CephFS 原生和 CephFS-NFS 被视为具有不同协议的后端。

共享文件系统服务的调度程序决定用于共享创建请求的目标后端。共享文件系统服务中的单一后端可以公开多个存储池。

当您配置多个后端时，调度程序会选择一个存储池，以从所有配置的后端公开的池中创建资源。这个过程是从最终用户中提取的。最终用户只能看到由云管理员公开的功能。

6.1.2. 部署多个后端

默认情况下，用于部署共享文件系统服务(manila)的环境文件只有一个后端。使用以下示例步骤来部署两个后端，即 CephFS-NFS 后端和 NetApp 后端。



重要

当您从同一供应商向部署添加多个存储后端时，请使用此流程配置一个后端。将其他后端配置为自定义后端。如需更多信息，请参阅配置自定义后端 和部署自定义后端。

先决条件

- 至少两个后端。
- 当您从需要外部软件组件的存储供应商中使用后端驱动程序时，您必须在部署过程中覆盖共享文件系统服务的标准容器镜像。例如，您可以在 红帽生态系统目录 中查找 Dell EMC unity 存储系统的 Dell EMC unity 容器镜像。

流程

1. 创建存储自定义 YAML 文件以提供适合您环境的任何值或覆盖：

```
$ vi /home/stack/templates/<multiple_backends>.yaml
```

   - 将 **<multiple_backends** > 替换为您的文件的名称。

2. 配置存储自定义 YAML 文件，使其包含任何覆盖并启用多个后端：

```
parameter_defaults:
  ManilaEnabledShareProtocols:
    - NFS
  ManilaNetappLogin: '<login_name>'
  ManilaNetappPassword: '<password>'
  ManilaNetappServerHostname: '<netapp-hostname>'
  ManilaNetappVserver: '<netapp-vserver>'
  ManilaNetappDriverHandlesShareServers: 'false'
```

   - 将尖括号 &lt **;&** gt; 中的值替换为 YAML 文件的正确值。

3. 以 **stack** 用户身份登录 undercloud 主机。

4. 查找 **stackrc** undercloud 凭证文件：

```
$ source ~/stackrc
```

5. 使用其他环境文件将存储自定义 YAML 文件添加到堆栈中，并部署 overcloud。这个示例配置启用带有 NetApp 后端和 CephFS-NFS 后端的共享文件系统服务。

```
$ openstack overcloud deploy \
  --timeout 100 \
  --stack overcloud \
  --templates /usr/share/openstack-tripleo-heat-templates \
  -n /usr/share/openstack-tripleo-heat-templates/network_data_ganesha.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-mds.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-
ansible.yaml \
  -r /home/stack/templates/roles/roles_data.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsganesha-
config.yaml \
```

```
-e /usr/share/openstack-tripleo-heat-templates/environments/manila-netapp-config.yaml \
-e /home/stack/templates/storage_customizations.yaml \
...
```

其他资源

- 有关 **ManilaEnabledShareProtocols** 参数的详情，请参考 第 6.1.7 节 “更改允许的 NAS 协议”。

6.1.3. 确认部署多个后端

使用 **manila service-list** 命令来验证您的共享文件系统服务(manila)后端是否已成功部署。如果您在多个后端中使用健康检查，ping 测试也会返回一个响应，即使其中一个后端不响应，因此这不是验证部署的一种可靠方法。

流程

1. 提供 overcloud 凭证文件：

```
$ source ~/<credentials_file>
```

   - 将 **<credentials_file** > 替换为您的凭据文件的名称，如 **overcloudrc**。

2. 确认共享文件系统服务后端列表：

```
$ manila service-list

+----+-------------+-------------+------+---------+-------+----------------------------+
| Id | Binary      | Host        | Zone | Status  | State | Updated_at                 |
+----+-------------+-------------+------+---------+-------+----------------------------+
| 2  | manila-scheduler | hostgroup | nova | enabled | up | 2021-03-24T16:49:09.000000 |
| 5  | manila-share | hostgroup@cephfs | nova | enabled | up | 2021-03-24T16:49:12.000000 |
| 8  | manila-share | hostgroup@tripleo_netapp | nova | enabled | up | 2021-03-
24T16:49:06.000000 |
```

   每个成功部署的后端的状态都显示为 **enabled**，并且状态显示为 **up**。

6.1.4. 配置自定义后端

您可以在 Red Hat OpenStack Platform (RHOSP) director 安装程序中部署没有对应实现的自定义存储后端。

当您从同一供应商向部署添加多个存储后端时，请使用 部署多个后端 中的步骤来配置一个后端。将其他后端配置为自定义后端。

当您部署自定义后端时，检查存储后端供应商是否需要自定义容器镜像，而不是共享文件系统服务(manila)的标准容器镜像。您可以在 红帽生态系统目录 中找到自定义容器镜像。

流程

1. 创建存储自定义 YAML 文件以提供适合您环境的任何值或覆盖：

```
$ vi /home/stack/templates/<custom_backend_data>.yaml
```

   - 将 **<custom_backend_data** > 替换为您的文件的名称。

2. 使用共享文件系统服务所需的容器镜像配置存储自定义 YAML 文件：

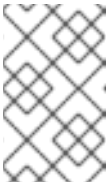
```
parameter_defaults:
  ContainerManilaShareImage: <custom_container_image_url>
```

- 将 **<custom_container_image_url** > 替换为自定义容器镜像的 URL。

3. 使用 **ControllerExtraConfig** 参数配置共享文件系统服务所需的后端。此参数确保配置应用到所有 Controller 节点：

```
parameter_defaults:
  ...
  ControllerExtraConfig:
    manila::config::manila_config:
      <backend_name>/<parameter>:
        value: '<parameter_value>'
```

- 将 **<backend_name** > 替换为自定义后端的名称。
- 将 **&lt**;parameter> 替换为您要为此后端配置的参数的名称，如 **netapp_server_hostname** 或 **netapp_password**。
- 将 **<parameter_value** > 替换为您要为参数设置的值，如 **203.0.113.20** 或 **admin_password**。



注意

如果使用自定义角色，请使用 **[role_name]ExtraConfig** 而不是 **ControllerExtraConfig** 参数。将 **[role_name]** 替换为您的自定义角色的名称。

6.1.5. 部署自定义后端

当您为 Red Hat OpenStack Platform (RHOSP)部署配置 自定义存储后端时，如 配置自定义后端 中所述，您可以将其添加到部署中。

流程

1. 创建存储自定义 YAML 文件以提供适合您环境的任何值或覆盖：

```
$ vi /home/stack/templates/<manila_enabled_share_backends>.yaml
```

- 将 **<manila_enabled_share_backends** > 替换为您的文件的名称。

2. 配置存储自定义 YAML 文件，将自定义后端添加到启用的共享后端列表中：

```
parameter_defaults:
  ControllerExtraConfig:
    manila_user_enabled_backends:
      - '<backend_name>'
```

- 将 **<backend_name** > 替换为自定义后端的名称。

3. 使用其他环境文件将存储自定义 YAML 文件添加到堆栈中，并部署 overcloud：

```
$ openstack overcloud deploy --templates \
-e [your environment files] \
  -e /home/stack/templates/<manila_enabled_share_backends>.yaml
```

6.1.6. 为后端部署可用区

您可以创建可用区(AZ)来对云用户的云基础架构和服务进行分组。您可以将 AZ 映射到故障域和计算资源，以实现高可用性、容错和资源调度。例如，您可以创建一个具有特定硬件的 Compute 节点的 AZ，该用户在创建需要该硬件的实例时可以指定这些硬件。

共享始终与 AZ 关联。如果您在创建共享时没有设置 AZ 参数，则共享文件系统服务(manila)将共享与名为 **nova** 的默认 AZ 关联。您可以使用 **ManilaXXXAvailabilityZone** 参数，其中 **XXX** 与特定后端关联，为共享文件系统服务后端配置不同的 AZ。有关 AZ 的更多信息，请参阅*配置计算服务以进行实例创建*中的创建和管理主机聚合。

重要

如果您更改了现有共享的 AZ 参数，则这些共享将继续与原始 AZ 关联，而其后端映射到新的 AZ。目前，当您更改现有共享上的 AZ 参数时，无法协调原始 AZ 和新映射的 AZ 参数之间的冲突。

流程

1. 在您的共享文件系统服务环境文件中添加以下参数，以创建两个 AZ：

```
parameter_defaults:
 ManilaXXXAvailabilityZone: zone1
 ManilaYYYAvailabilityZone: zone2
```

   - 将 **XXX** 和 **YYY** 替换为支持的后端值，例如：

```
ManilaCephFSAvailabilityZone
ManilaNetAppAvailabilityZone
```

     以下示例部署两个后端，其中 **CephFS** 是区 1，**NetApp** 是 zone 2：

```
parameter_defaults:
 ManilaCephFSAvailabilityZone: zone1
 ManilaNetAppAvailabilityZone: zone2
```

     注意

     在 **/usr/share/openstack-tripleo-heat-templates/deployment/manila/** 目录中搜索与后端关联的 heat 模板，以获取正确的后端值。

2. 以 **stack** 用户身份登录 undercloud 主机。

3. 查找 **stackrc** undercloud 凭证文件：

```
$ source ~/stackrc
```

4. 使用其他环境文件将更新的共享文件系统服务环境文件添加到堆栈中，并部署 overcloud：

```
$ openstack overcloud deploy --templates \
  -e [your environment files] \
  -e /home/stack/templates/<updated_environment_file>.yaml
```

5. 部署后，使用 **availability_zones** 共享类型额外规格将共享类型限制为一个或多个 AZ。只要共享类型没有限制，云用户可以直接在 AZ 中创建共享。

其他资源

- 创建共享类型
- 在 *强化 Red Hat OpenStack Platform* 时 共享类型访问控制。

6.1.7. 更改允许的 NAS 协议

您可以使用共享文件系统服务(manila)在 NFS、CephFS 或 CIFS 网络附加存储(NAS)协议中导出共享。默认情况下，共享文件系统服务启用部署中后端支持的所有 NAS 协议。

作为 Red Hat OpenStack Platform (RHOSP)管理员，您可以更改 **ManilaEnabledShareProtocols** 参数，并只列出您要在云中允许的协议。例如，如果您的部署中的后端同时支持 NFS 和 CIFS，您可以更改默认值并启用一个协议。

并非所有存储后端驱动程序都支持 CIFS 协议。有关哪些认证存储系统支持 CIFS 的详情，请查看 红帽生态系统目录。有关通过 RHOSP director 配置服务的详情，请参考厂商存储文档。

流程

1. 提供 overcloud 凭证文件：

```
$ source ~/<credentials_file>
```

   - 将 **<credentials_file** > 替换为您的凭据文件的名称，如 **overcloudrc**。

2. 创建存储自定义 YAML 文件以提供适合您环境的任何值或覆盖：

```
$ vi /home/stack/templates/<share_protocols>.yaml
```

   - 将 **<share_protocols** > 替换为您的文件的名称。

3. 使用您要启用的 NAS 协议配置 **ManilaEnabledShareProtocols** 参数：

```
parameter_defaults:
  ManilaEnabledShareProtocols:
    - NFS
    - CEPHFS
```

4. 使用其他环境文件将存储自定义 YAML 文件添加到堆栈中，并部署 overcloud：

```
$ openstack overcloud deploy --templates \
-e [your environment files] \
  -e /home/stack/templates/<share_protocols>.yaml
```

注意

部署不会验证设置。您分配的 NAS 协议必须被共享文件系统服务部署中的后端支持。

6.1.8. 查看后端功能

共享文件系统服务(manila)的调度程序组件根据几个因素（如容量、配置配置、放置提示以及后端存储系统驱动程序检测到和公开的功能）做出智能放置决策。

流程

- 运行以下命令来查看可用功能：

```
$ manila pool-list --detail
+-----------------------------------+---------------------------+
| Property                          | Value                     |
+-----------------------------------+---------------------------+
| name                              | hostgroup@cephfs#cephfs    |
| pool_name                         | cephfs                    |
| total_capacity_gb                 | 1978                      |
| free_capacity_gb                  | 1812                      |
...
| driver_handles_share_servers      | False                     |
| snapshot_support                  | True                      |
| create_share_from_snapshot_support | False                    |
| revert_to_snapshot_support        | False                     |
| mount_snapshot_support            | False                     |
...
+-----------------------------------+---------------------------+
+-----------------------------------+---------------------------------+
| Property                          | Value                           |
+-----------------------------------+---------------------------------+
| name                              | hostgroup@tripleo_netapp#aggr1_n1 |
| pool_name                         | aggr1_n1                        |
| total_capacity_gb                 | 6342.1                          |
| free_capacity_gb                  | 6161.99                         |
...
| driver_handles_share_servers      | False                           |
| mount_snapshot_support            | False                           |
| replication_type                  | None                            |
| replication_domain                | None                            |
| sg_consistent_snapshot_support    | host                            |
| ipv4_support                      | True                            |
| ipv6_support                      | False                           |
+-----------------------------------+---------------------------------+
+-----------------------------------+---------------------------------+
| Property                          | Value                           |
+-----------------------------------+---------------------------------+
| name                              | hostgroup@tripleo_netapp#aggr1_n2 |
| pool_name                         | aggr1_n2                        |
| total_capacity_gb                 | 6342.1                          |
| free_capacity_gb                  | 6209.26                         |
...
| snapshot_support                  | True                            |
| create_share_from_snapshot_support | True                           |
| revert_to_snapshot_support        | True                            |
| driver_handles_share_servers      | False                           |
| mount_snapshot_support            | False                           |
| replication_type                  | None                            |
| replication_domain                | None                            |
```

```
| sg_consistent_snapshot_support    | host                                |
| ipv4_support                      | True                                |
| ipv6_support                      | False                               |
+-----------------------------------+-------------------------------------+
```

相关信息

要影响放置决策，作为管理员，您可以使用共享类型和额外规格。有关共享类型的更多信息，请参阅创建共享类型。

6.2. 创建共享类型

云管理员可以创建共享类型来定义共享文件系统服务(manila)调度程序用来做出调度决策和驱动程序用来控制共享创建的服务类型。

共享类型包括描述和额外规格，例如 **driver_handles_share_servers** 和 **snapshot_support** 来过滤后端。Red Hat OpenStack Platform (RHOSP) director 使用名为 **default** 的默认共享类型配置共享文件系统服务，但 director 不会创建共享类型。

云用户需要至少一个共享类型才能使用共享文件系统服务，用户只能创建与可用共享类型匹配的共享。

默认情况下，共享类型是公共的，这意味着它们适用于所有云项目。但是，您可以创建私有共享类型，以便在特定项目中使用。

在以下示例中，您可以使用 **driver_handles_share_servers** 参数(DHSS)，它可以设置为 **true** 或 **false**：

- 对于 CephFS-NFS 和原生 CephFS，您可以将 DHSS 设置为 **false**。
- 对于其他后端，您可以将 DHSS 设置为 **true** 或 **false**。您可以将 DHSS 值设置为与存储自定义环境文件中的 **Manila<backend>DriverHandlesShareServers** 参数的值匹配。例如，如果您使用 NetApp 后端，则此参数是 **ManilaNetappDriverHandlesShareServers**。

流程

1. 部署 overcloud 后，运行以下命令来创建共享类型：

```
$ manila type-create default <driver_handles_share_servers>
```

   - 将 **<driver_handles_share_servers>** 替换为 **true** 或 **false**。

2. 向默认共享类型添加规格，或创建额外的共享类型以用于不同的后端。在本例中，将默认共享类型配置为选择 CephFS 后端，并使用 NetApp **driver_handles_share_servers=true** 后端的额外共享类型：

```
$ manila type-create default false \
  --extra-specs share_backend_name='cephfs'
$ manila type-create netapp true \
  --extra-specs share_backend_name='tripleo_netapp'
```

其他资源

- 有关如何制作私有共享类型或设置其他共享类型选项的更多信息，请参阅 强化 Red Hat OpenStack Platform。

6.3. 比较共享类型的常见功能

共享类型定义了共享的常见功能。查看共享类型的常见功能，以了解您可以与共享相关的内容。

表 6.1. 共享类型的功能

功能	值	描述
driver_handles_share_servers	true 或 false	授予使用共享网络创建共享的权限。
snapshot_support	true 或 false	授予创建共享快照的权限。
create_share_from_snapshot_support	true 或 false	授予创建共享快照克隆的权限。
revert_to_snapshot_support	true 或 false	授予将共享恢复到最新快照的权限。
mount_snapshot_support	true 或 false	授予导出和挂载快照的权限。
replication_type	dr	授予为灾难恢复创建副本的权限。一次只允许一个活跃的导出。
	可读	授予创建只读副本的权限。一次只允许一个可写的、活跃的导出。
	writable	授予创建读/写副本的权限。每个共享允许任意数量活跃的导出。
availability_zones	一个或多个可用区的列表	授予仅在列出的可用区上创建共享的权限。

6.4. 使用 MANAGE/UNMANAGE添加和删除共享

云管理员可以通过共享文件系统服务(manila)管理/取消管理功能来管理存储中已存在的文件共享。您可以对受管共享执行操作，如授予访问权限、挂载和调整大小，这与在共享文件系统服务共享上执行这些操作的方式相同。

您可以管理将 **driver_handles_share_servers** 参数(DHSS)设置为 true 的共享的生命周期，并将 DHSS 设置为 false 的共享。要管理 DHSS=true 共享，云管理员还必须管理包含该共享的共享服务器。

取消管理共享时，您可以在不删除共享的情况下从共享文件系统服务的管理中删除共享。如果共享有依赖的快照或共享副本，则只能在快照或共享副本被删除时从共享文件系统服务中删除共享。

限制

- 驱动程序必须支持 manage/unmanage 功能。

- manage/unmanage 功能不支持原生 CephFS 或 CephFS-NFS 后端。您可以从共享文件系统服务的管理中删除 CephFS 共享。但是，您无法将现有的 CephFS 共享置于共享文件系统服务的管理下。
- 当您在管理共享文件系统服务下建立共享时，现有客户端会断开连接。当您从共享文件系统服务的管理中删除共享时，现有客户端将保持连接。

流程

1. 管理共享：

```
manila manage
--name <name>
--description <description>
--share_type <share-type>
--driver_options [<key=value> [<key=value> ...]]
                [--public] [--share_server_id <share-server-id>] \
                [--wait] <service_host> <protocol> <export_path>
```

   - 将尖括号 &lt **;&** gt; 中的值替换为您的环境的正确值。

2. 验证共享是否可用：

```
$ manila show <name>
```

3. 取消管理共享：

```
manila unmanage [--wait] <name>
```

6.5. 为共享文件系统规划网络

规划云上的网络，以确保最终用户客户端将其共享连接到在 Red Hat OpenStack Platform (RHOSP)虚拟机、裸机服务器和容器上运行的工作负载。

根据云用户所需的安全性和隔离级别，您可以将 **driver_handles_share_servers** 参数(DHSS)设置为 **true** 或 **false**。

DHSS=true

如果将 DHSS 参数设置为 **true**，您可以使用共享文件系统服务(manila)将共享导出到具有隔离共享服务器的最终用户定义的共享网络。用户可以在自助服务共享网络上配置工作负载，以确保在专用网络片段上隔离 NAS 文件服务器导出其共享。

作为云管理员，您必须确保将隔离网络映射到您的存储基础架构的物理网络。您还必须确保您使用的存储系统支持网络段。存储系统，如 NetApp ONTAP 和 Dell EMC PowerMax、Unity 和 VNX，不支持 GENEVE 或 VXLAN 等虚拟覆盖分段样式。

作为覆盖网络的替代方案，您可以执行以下操作之一：

- 将 VLAN 网络用于您的项目网络。
- 允许共享提供商网络上的 VLAN 片段。
- 提供对已连接到您的存储系统的预先存在的网段网络的访问。

DHSS=false

如果将 DHSS 参数设置为 **false**，则云用户无法在自己的共享网络上创建共享。它们必须将自己的客户端连接到云管理员所配置的网络。

作为云管理员，您可以通过 director 创建专用共享存储网络。例如，当使用标准 director 模板部署原生 CephFS 后端时，您将生成一个名为 **Storage** 的共享提供商网络。当您通过 NFS 后端部署 CephFS 时，您将生成一个名为 **StorageNFS** 的共享提供商网络。云用户必须将其客户端连接到共享存储网络，才能访问其共享。

并非所有共享文件系统存储驱动程序都支持 DHSS=true 和 DHSS=false。DHSS=true 和 DHSS=false 可确保数据路径多租户隔离。但是，如果您需要租户工作负载的网络路径多租户隔离，作为自助服务模型的一部分，则必须使用支持 DHSS=true 的后端部署共享文件系统服务。

有关与共享的网络连接的详情，请参考 第 6.6 节 “确保到共享的网络连接”。

6.6. 确保到共享的网络连接

要连接到文件共享，客户端必须具有与该共享的一个或多个导出位置的网络连接。

当云管理员将共享类型的 **driver_handles_share_servers** 参数设置为 true 时，云用户可以使用 Compute 实例附加的网络详情创建共享网络。云用户在创建共享时可以引用共享网络。

当云管理员将共享类型设置为 false 时，云用户必须将其 Compute 实例连接到云管理员配置的共享存储网络。有关如何配置并验证到共享网络的网络连接的详情，请参考 第 7.5 节 “连接到共享网络以访问共享共享”。

6.7. 更改共享文件系统服务中的默认配额

为防止系统容量在没有通知的情况下耗尽，云管理员可以配置配额。配额是操作限制。共享文件系统服务(manila)默认强制执行一些可行的限制。这些限制称为默认配额。云管理员可覆盖默认配额，以便单个项目具有不同的消耗限制。

6.7.1. 更新项目、用户和共享类型的配额

作为云管理员，您可以使用 **manila quota-show** 命令列出项目或用户的配额。

您可以为项目中的所有用户或特定项目用户更新配额，或者项目用户使用的共享类型。您可以为您选择的目标更新以下配额：

- **共享** ：您可以创建的共享数。
- **快照** ：您可以创建的快照数。
- **gigabytes** ：您可以为所有共享分配的总大小（以 GB 为单位）。
- **snapshot-gigabytes** ：您可以为所有共享快照分配的总大小（以 GB 为单位）。
- **share-networks** ：您可以创建的共享网络总数。
- **share_groups** ：您可以创建的共享组总数。
- **share_group_snapshots** ：您可以创建的共享组快照总数。
- **share-replicas** ：您可以创建的共享副本数。
- **replica-gigabytes** ：您可以在所有共享副本间分配的总大小（以 GB 为单位）。

注意

您只能在项目级别指定 **共享类型** 配额。您不能 为**特定项目用户设置共享类型** 配额。

重要

在以下步骤中，仔细输入值。共享文件系统服务不会检测或报告不正确的值。

流程

1. 您可以使用以下命令来查看配额。如果包含 **--user** 选项，您可以在指定项目中查看特定用户的配额。如果省略 **--user** 选项，您可以查看应用到指定项目的所有用户的配额。
同样，如果您包含可选的 **--share-type**，您可以查看特定共享类型的配额，因为它与项目相关。**--user** 和 **--share-type** 选项是互斥的。

```
$ manila quota-show
```

- 项目示例：

```
$ manila quota-show \
  --project af2838436f3f4cf6896399dd97c4c050
+----------------------+----------------------------------+
| Property             | Value                            |
+----------------------+----------------------------------+
| gigabytes            | 1000                             |
| id                   | af2838436f3f4cf6896399dd97c4c050 |
| replica_gigabytes    | 1000                             |
| share_group_snapshots | 50                              |
| share_groups         | 49                               |
| share_networks       | 10                               |
| share_replicas       | 100                              |
| shares               | 50                               |
| snapshot_gigabytes   | 1000                             |
| snapshots            | 50                               |
+----------------------+----------------------------------+
```

- 项目用户示例：

```
$ manila quota-show \
  --project af2838436f3f4cf6896399dd97c4c050 \
  --user 81ebb491dd0e4c2aae0775dd564e76d1
+----------------------+----------------------------------+
| Property             | Value                            |
+----------------------+----------------------------------+
| gigabytes            | 500                              |
| id                   | af2838436f3f4cf6896399dd97c4c050 |
| replica_gigabytes    | 1000                             |
| share_group_snapshots | 50                              |
| share_groups         | 49                               |
| share_networks       | 10                               |
| share_replicas       | 100                              |
| shares               | 25                               |
| snapshot_gigabytes   | 1000                             |
| snapshots            | 50                               |
+----------------------+----------------------------------+
```

- 特定共享类型的项目示例：

```
$ manila quota-show \
  --project af2838436f3f4cf6896399dd97c4c050 \
  --share-type dhss_false
+--------------------+----------------------------------+
| Property           | Value                            |
+--------------------+----------------------------------+
| gigabytes          | 1000                             |
| id                 | af2838436f3f4cf6896399dd97c4c050 |
| replica_gigabytes  | 1000                             |
| share_replicas     | 100                              |
| shares             | 15                               |
| snapshot_gigabytes | 1000                             |
| snapshots          | 50                               |
+--------------------+----------------------------------+
```

2. 使用 **manila quota-update** 命令更新配额。您可以为所有项目用户、特定项目用户或项目中的共享类型更新配额：

- 为项目中的所有用户更新配额：

```
$ manila quota-update <id> \
  [--shares <share_quota> \
  --gigabytes <share_gigabytes_quota> \
  …]
```

将 & **lt;id&** gt; 替换为项目 ID。这个值必须是项目 ID，而不是项目名称。

- 为项目中的特定用户更新配额：

```
$ manila quota-update <id> \
  --user <user_id> \
  [--shares <new_share_quota> \
  --gigabytes <new_share_gigabytes_quota> \
  …]
```

  - 将 & **lt;id&** gt; 替换为项目 ID。这个值必须是项目 ID，而不是项目名称。
  - 将 **<user_id&** gt; 替换为用户 ID。该值必须是用户 ID，而不是用户名。

- 为使用特定共享类型的所有用户更新配额：

```
$ manila quota-update <id> \
  --share-type <share_type> \
  [--shares <new_share_quota>30
  --gigabytes <new-share_gigabytes_quota> \
  …]
```

  - 将 & **lt;id&** gt; 替换为项目 ID。这个值必须是项目 ID，而不是项目名称。
  - 将 < **share_type** > 替换为您要将配额应用到的共享类型的名称或 ID。

验证

- **quota-update** 命令不会生成任何输出。使用 **quota-show** 命令验证配额是否已成功更新。

6.7.2. 为项目、用户和共享类型重置配额

您可以删除配额覆盖，将配额返回到默认值。目标实体受默认配额限制，适用于所有没有覆盖的项目。

流程

- 使用 **manila quota-delete** 命令将配额返回到默认值。您可以将配额返回到所有项目用户、特定项目用户或项目中的共享类型的默认值：
  - 重置项目配额：

    ```
    $ manila quota-delete --project <id>
    ```

    将 &**lt;id&** gt; 替换为项目 ID。这个值必须是项目 ID，而不是项目名称。
  - 为特定用户重置配额：

    ```
    $ manila quota-delete --project <id> --user <user_id>
    ```

    - 将 &**lt;id&** gt; 替换为项目 ID。这个值必须是项目 ID，而不是项目名称。
    - 将 **<user_id&** gt; 替换为用户 ID。该值必须是用户 ID，而不是用户名。
  - 为项目用户使用的共享类型重置配额：

    ```
    $ manila quota-delete --project <id> --share-type <share_type>
    ```

    - 将 &**lt;id&** gt; 替换为项目 ID。这个值必须是项目 ID，而不是项目名称。
    - 将 **<share_type** > 替换为必须重置配额的共享类型的名称或 ID。

验证

1. **quota-delete** 命令不会生成任何输出。使用 **quota-show** 命令来验证配额是否已成功重置。
2. 列出所有项目的默认配额。默认配额应用到没有覆盖的项目。

   ```
   $ manila quota-class-show default
   ```

6.7.3. 更新共享文件系统服务项目的默认配额值

作为云管理员，您可以更新应用到尚未有配额覆盖的所有项目的默认配额。

流程

1. 查看 **manila quota-class-update** 命令的 usage 语句：

   ```
   $ manila help quota-class-update
   usage: manila quota-class-update [--shares <shares>] [--snapshots <snapshots>]
                                    [--gigabytes <gigabytes>]
                                    [--snapshot-gigabytes <snapshot_gigabytes>]
                                    [--share-networks <share_networks>]
   ```

```
[--share-replicas <share_replicas>]
[--replica-gigabytes <replica_gigabytes>]
<class_name>
```

注意

参数 & **lt;class_name** > 是一个位置参数。它标识设置配额的配额类。将此参数设置为 **default**。不支持其他配额类。

您可以为以下任何可选参数更新值：

- **--shares <shares** > 为 **共享** 配额添加新值。
- **--snapshots <snapshots** > 为 **快照** 配额添加新值。
- **--gigabytes <gigabytes>** 添加一个新的 **gigabytes** 配额。
- **--snapshot-gigabytes <snapshot_gigabytes&** gt **; 或 --snapshot_gigabytes <snapshot_gigabytes** > 为 **snapshot_gigabytes** 配额添加一个新值。
- **--share-networks <share_networks&** gt **; 或 --share_networks <share_networks** > 为 **share_networks** 配额添加一个新值。
- **--share-replicas <share_replicas&** gt **; , --share_replicas <share_replicas** &gt **; 或 --replicas <share_replicas** > 为 **share_replicas** 配额添加一个新的值。
- **--replica-gigabytes <replica_gigabytes&** gt **; 或 --replica_gigabytes <replica_gigabytes** > 为 **replica_gigabytes** 配额添加一个新的值。

2. 使用 usage 语句中的信息来更新默认配额。以下示例为 **shares** 和 **gigabytes** 更新默认配额：

```
$ manila quota-class-update default \
  --shares 30 \
  --gigabytes 512
$ manila quota-class-show default
+----------------------+---------+
| Property             | Value   |
+----------------------+---------+
| gigabytes            | 512     |
| id                   | default |
| replica_gigabytes    | 1000    |
| share_group_snapshots | 50     |
| share_groups         | 50      |
| share_networks       | 10      |
| share_replicas       | 100     |
| shares               | 30      |
| snapshot_gigabytes   | 1000    |
| snapshots            | 50      |
+----------------------+---------+
```

第 7 章 使用共享文件系统服务(MANILA)执行操作

云用户可以通过共享文件系统服务(manila)中的可用共享类型创建和管理共享。您可以使用共享文件系统服务命令行客户端来管理共享文件系统。

7.1. 列出共享类型

作为云用户，在创建共享时必须指定共享类型。必须至少有一个可用的共享类型才能使用共享文件系统服务(manila)，您只能创建与可用共享类型匹配的共享。云管理员配置共享类型，以定义共享文件系统服务调度程序用来做出调度决策和驱动程序用来控制共享创建的服务类型。

流程

- 列出可用的共享类型：

```
$ manila type-list
```

命令输出列出了可用共享类型的名称和 ID。

7.2. 创建 NFS、CEPHFS 或 CIFS 共享

云用户可以创建 CephFS-NFS、原生 CephFS 或 CIFS 共享来读取和写入数据。

创建共享时，您必须以 GB 为单位指定共享协议和共享大小。您还可以包含 **share-type**、**share-network** 和 **name** 命令选项：

```
$ manila create [--share-type <share_type>] \
    [--share-network <share_network>]  \
    [--name <share_name>] <share_protocol> <GB>
```

在命令示例中，替换以下值：

- **<share_type>**：应用与指定共享类型关联的设置：
  - 可选。如果没有指定共享类型，则使用默认共享类型。
- **<share_network>**：共享网络的名称：
  - **如果共享类型将 driver_handles_share_servers 设置为 true，则需要此项。**
  - **如果共享类型将 driver_handles_share_servers 设置为 false，则不支持。**
  - **CephFS-NFS 和原生 CephFS 不支持。这些协议不支持将 driver_handles_share_servers 设置为 true 的共享类型。**

- <share_name >：共享名称：
  - 可选。共享不需要具有名称，且名称不需要唯一。
- <share_protocol > ：您要使用的共享协议：
  - 对于 CephFS-NFS，将 &lt ;share_protocol> 替换为 nfs。
  - 对于原生 CephFS，将 &lt ;share_protocol> 替换为 cephfs。
  - 对于支持 NFS 或 CIFS 协议的其他存储后端，例如 NetApp 或 Dell EMC 存储后端，将 < share_protocol > 替换为 nfs 或 cifs。
- <GB > ：共享的大小（以 GB 为单位）。

7.2.1. 使用 DHSS=true 创建 NFS 或 CIFS 共享

当云管理员使用共享类型额外规格 driver_handles_share_servers=true 激活自助服务共享网络时，云用户可以将自己的安全服务添加到共享网络中，以创建并导出 NFS 或 CIFS 共享。原生 CephFS 协议不支持共享网络。

要添加安全服务，您必须创建一个共享网络和安全服务资源来代表您的活动目录服务器。然后，您可以将安全服务与共享网络关联，以创建并导出 NFS 或 CIFS 共享。

流程

1. 创建共享网络：

```
$ manila share-network-create --name <network-name> \
  --neutron-net-id <25d1e65c-d961-4f22-9476-1190f55f118f> \
  --neutron-subnet-id <8ba20dce-0ca5-4efd-bf1c-608d6bceffe1>
```

   - 将 < network-name > 替换为您要用于 NFS 或 CIFS 共享的共享网络名称。

- 将 neutron-net-id 和 neutron-subnet-id 替换为您的共享网络的正确值。

2. 创建一个安全服务资源来代表您的活动目录服务器：

```
$ manila security-service-create <active_directory> \
        --dns-ip <192.02.12.10> \
        --domain <domain-name.com> \
        --user <Administrator> \
        --password <password> \
        --name <AD-service>
```

- 将尖括号 &lt ;& gt; 中的值替换为安全服务资源的正确详情。

3. 将安全服务资源与共享网络关联：

```
$ manila share-network-security-service-add \
 <network-name> <AD-service>
```

4. 创建 NFS 或 CIFS 共享：

- 10 GB NFS 示例：

```
$ manila create --name <nfs-share> --share-type <netapp> \
   --share-network <nfs-network> nfs 10
```

- 20 GB CIFS 示例：

```
$ manila create --name <cifs-share> --share-type dhss_true \
   --share-network <cifs-network> cifs 20
```

  - 将尖括号 <& gt; 中的值替换为您的 NFS 或 CIFS 共享的正确详情。

7.2.2. 使用 DHSS=false 创建 NFS、CephFS 或 CIFS 共享

当云管理员使用共享类型额外规格 driver_handles_share_servers=false 停用自助服务共享网络时，它们必须预配置 Active Directory 服务到存储系统。有关如何执行此配置的详情，请查看存储厂商的文档。

当 DHSS=false 时，共享存储网络由云管理员预配置，云用户可以在不使用 share-network 命令选项的情况下创建共享。

流程

- 当 DHSS=false 时，创建 NFS、原生 CephFS 或 CIFS 共享。这些示例 指定名称，但不指定 share-type 或 share-network。它们 使用默认的 共享类型和云管理员配置的共享存储网络：
  - 创建一个名为 **share-01 的 10 GB NFS 共享。**

```
$ manila create --name share-01 nfs 10
```

  - 创建名为 **share-02 的 15 GB 原生 CephFS 共享：**

```
$ manila create --name share-02 cephfs 15
```

  - 创建名为 **share-03 的 20 GB CIFS 共享：**

```
$ manila create --name share-03 cifs 20
```

7.3. 列出共享和导出信息

要验证您在共享文件系统服务(manila)中成功创建了 NFS、CephFS 或 CIFS 共享，请完成以下步骤以列出共享并查看其导出位置和参数。

流程

1. 列出共享：

```
$ manila list

+--------------------------------------+----------+-----+-----------+
| ID                                   | Name     | ... | Status    ...
+--------------------------------------+----------+-----+-----------+
| 8c3bedd8-bc82-4100-a65d-53ec51b5fe81 | share-01 | ... | available ...
+--------------------------------------+----------+-----+-----------+
```

2. 查看共享的导出位置：

```
$ manila share-export-location-list <share>

+-----------------------------------------------------------------
| Path
|  198.51.100.13:/volumes/_nogroup/e840b4ae-6a04-49ee-9d6e-67d4999fbc01
+-----------------------------------------------------------------
```

- 将 <share > 替换为共享名称或共享 ID。

3. 查看共享的参数：

```
$ manila share-export-location-show <share-id>
```

注意

您可以使用导出位置挂载共享，如 第 7.8.2 节 “挂载 NFS、原生 CephFS 或 CIFS 共享” 所述。

7.4. 在共享文件系统中创建数据快照

快照是共享上数据的只读点复制。您可以使用快照恢复通过意外删除或文件系统损坏丢失的数据。快照比备份更高效，不会影响共享文件系统服务(manila)的性能。

先决条件

- 在父共享上，snapshot_support 参数必须是 true。您可以运行以下命令来验证：

```
$ manila show | grep snapshot_support
```

流程

1. 作为云用户，创建共享的快照：

```
$ manila snapshot-create [--name <snapshot_name>] <share>
```

- 将 & lt;share > 替换为您要为其创建快照的共享的名称或 ID。
- 可选：将 <snapshot_name > 替换为快照的名称。

输出示例

```
+-------------------+--------------------------------------+
| Property          | Value                                |
+-------------------+--------------------------------------+
| id                | dbdcb91b-82ba-407e-a23d-44ffca4da04c |
| share_id          | ee7059aa-5887-4b87-b03e-d4f0c27ed735 |
| share_size        | 1                                    |
| created_at        | 2022-01-07T14:20:55.541084           |
| status            | creating                             |
| name              | snapshot_name                        |
| description       | None                                 |
| size              | 1                                    |
| share_proto       | NFS                                  |
| provider_location | None                                 |
| user_id           | 6d414c62237841dcbe63d3707c1cdd90     |
| project_id        | 041ff9e24eba469491d770ad8666682d     |
+-------------------+--------------------------------------+
```

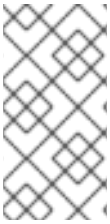
2. 确认您创建了快照：

```
$ manila snapshot-list --share-id <share>
```

将 & lt;share > 替换为您创建快照共享的名称或 ID。

7.4.1. 从快照创建共享

您可以从快照创建共享。如果从 创建了快照的父共享，其共享类型为 driver_handles_share_servers 设为 true，则在与父级相同的共享网络上创建新的共享。



注意

如果父共享的共享类型将 **driver_handles_share_servers** 设置为 **true**，则无法更改您从快照创建的共享的共享网络。

先决条件

- **create_share_from_snapshot_support** 共享属性设为 **true**。

  有关共享类型的更多信息，请参阅共享类型的通用功能。

- 快照的 **status** 属性设置为 **available**。

流程

1. 检索包含新共享所需数据的共享快照的 **ID**：

```
$ manila snapshot-list
```

2. 从快照创建的共享可能会更大，但不能小于快照。检索快照的大小：

```
$ manila snapshot-show <snapshot-id>
```

3. 从快照创建共享：

```
$ manila create <share_protocol> <size> \
  --snapshot-id <snapshot_id> \
  --name <name>
```

   - 将 **<share_protocol >** 替换为协议，如 **NFS**。
   - 将 **< size>** 替换为要创建的共享的大小，以 **GiB** 为单位。
   - 将 **<snapshot_id >** 替换为快照的 **ID**。

- 将 <name > 替换为新共享的名称。

4. 列出共享，以确认共享是否已成功创建：

```
$ manila list
```

5. 查看新共享的属性：

```
$ manila show <name>
```

验证

创建快照后，确认快照可用。

- 列出快照以确认它们可用：

```
$ manila snapshot-list
```

7.4.2. 删除快照

当您为共享创建快照时，在从该共享中删除创建的所有快照前，您无法删除共享。

流程

1. 识别您要删除的快照并检索其 ID：

```
$ manila snapshot-list
```

2. 删除快照：

```
$ manila snapshot-delete <snapshot>
```

注意

对您要删除的每个快照重复此步骤。

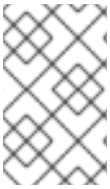
3. 删除快照后，运行以下命令确认已删除快照：

```
$ manila snapshot-list
```

7.5. 连接到共享网络以访问共享共享

当 driver_handles_share_servers 参数(DHSS)等于 false 时，共享将导出到云管理员可用的共享提供商网络。作为最终用户，您必须将您的客户端（如 Compute 实例）连接到共享提供商网络，以访问您的共享。

在本例中，共享供应商网络称为 StorageNFS。当 director 使用 CephFS-NFS 后端部署共享文件系统服务(manila)时，会配置 StorageNFS。按照以下步骤连接到您的云管理员可用的网络。



注意

示例流程中的步骤使用 IPv4 寻址，但步骤与 IPv6 相同。

流程

1. 为 StorageNFS 端口创建一个安全组，它允许数据包出站端口，但不允许来自未建立的连接的入口数据包：

```
$ openstack security group create no-ingress -f yaml
created_at: '2018-09-19T08:19:58Z'
description: no-ingress
id: 66f67c24-cd8b-45e2-b60f-9eaedc79e3c5
name: no-ingress
project_id: 1e021e8b322a40968484e1af538b8b63
revision_number: 2
rules: 'created_at="2018-09-19T08:19:58Z", direction="egress", ethertype="IPv4",
 id="6c7f643f-3715-4df5-9fef-0850fb6eaaf2", updated_at="2018-09-19T08:19:58Z"
 created_at="2018-09-19T08:19:58Z", direction="egress", ethertype="IPv6",
 id="a8ca1ac2-fbe5-40e9-ab67-3e55b7a8632a", updated_at="2018-09-19T08:19:58Z"
```

2. 在 StorageNFS 网络中创建一个端口，其安全性由 no-ingress 安全组强制使用。

```
$ openstack port create nfs-port0 \
  --network StorageNFS \
  --security-group no-ingress -f yaml

admin_state_up: UP
allowed_address_pairs: ''
```

```
binding_host_id: null
binding_profile: null
binding_vif_details: null
binding_vif_type: null
binding_vnic_type: normal
created_at: '2018-09-19T08:03:02Z'
data_plane_status: null
description: ''
device_id: ''
device_owner: ''
dns_assignment: null
dns_name: null
extra_dhcp_opts: ''
fixed_ips: ip_address='198.51.100.160', subnet_id='7bc188ae-aab3-425b-a894-
863e4b664192'
id: 7a91cbbc-8821-4d20-a24c-99c07178e5f7
ip_address: null
mac_address: fa:16:3e:be:41:6f
name: nfs-port0
network_id: cb2cbc5f-ea92-4c2d-beb8-d9b10e10efae
option_name: null
option_value: null
port_security_enabled: true
project_id: 1e021e8b322a40968484e1af538b8b63
qos_policy_id: null
revision_number: 6
security_group_ids: 66f67c24-cd8b-45e2-b60f-9eaedc79e3c5
status: DOWN
subnet_id: null
tags: ''
trunk_details: null
updated_at: '2018-09-19T08:03:03Z'
```

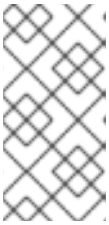
注意

在本例中，StorageNFS 网络上的 StorageNFS 子网将 IP 地址 198.51.100.160 分配给 nfs-port0。如需有关 StorageNFS 子网的更多信息，请参阅 *部署 Red Hat Ceph Storage 和 Red Hat OpenStack Platform* 中的 配置共享供应商 StorageNFS 网络。

3. **将 nfs-port0 添加到 Compute 实例。**

```
$ openstack server add port instance0 nfs-port0
$ openstack server list -f yaml
- Flavor: m1.micro
  ID: 0b878c11-e791-434b-ab63-274ecfc957e8
  Image: manila-test
  Name: demo-instance0
  Networks: demo-network=198.51.100.4, 10.0.0.53; StorageNFS=198.51.100.160
  Status: ACTIVE
```

除了其私有和浮动地址外，计算实例还分配有 StorageNFS 网络上的 IP 地址 198.51.100.160 的端口。当将访问权限授予该共享的地址时，您可以使用此 IP 地址挂载 NFS 共享。



注意

您可能需要调整 Compute 实例上的网络配置，并重启 Compute 实例的服务来激活使用此地址的接口。

7.6. 在网络和实例之间配置 IPV6 接口

当导出共享的共享网络使用 IPv6 寻址时，您可能会在二级接口上遇到 DHCPv6 的问题。如果出现这个问题，请在实例上手动配置 IPv6 接口。

先决条件

- 连接到共享网络以访问共享共享

流程

1. 登录实例。
2. 配置 IPv6 接口地址：

```
$ sudo ip address add fd00:fd00:fd00:7000::c/64 dev eth1
```

3. 激活接口：

```
$ sudo ip link set dev eth1 up
```

4. 在共享的导出位置 ping IPv6 地址来测试接口连接：

```
$ ping -6  fd00:fd00:fd00:7000::21
```

5. 或者，验证您可以通过 Telnet 访问 NFS 服务器：

```
$ sudo dnf install -y telnet
$ telnet fd00:fd00:fd00:7000::21 2049
```

7.7. 为最终用户客户端授予共享访问权限

您必须向最终用户客户端授予对共享的访问权限，以便用户可以从共享中读取数据并将数据写入共享。

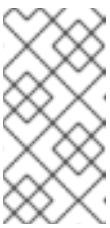
您可以通过实例的 IP 地址授予客户端计算实例对 NFS 共享的访问权限。CIFS 共享的 user 规则和 CephFS 共享的 cephx 规则有类似的特征。对于 用户和 cephx 访问类型，如果需要，您可以在 多个客户端间使用相同的客户端标识符。

在客户端（如计算实例）上挂载共享前，您必须使用类似如下的命令授予客户端对共享的访问权限：

```
$ manila access-allow <share> <accesstype> \
  --access-level <accesslevel>  <clientidentifier>
```

替换以下值：

- 共享 ：创建的共享的共享名称或 ID，如 第 7.2 节 “创建 NFS、CephFS 或 CIFS 共享” 所述。
- accesstype ：在共享中请求的访问权限类型。有些类型包括：
  - 用户 ：使用 按用户或组名称进行身份验证。
  - IP ：使用 通过其 IP 地址对实例进行身份验证。
  - cephx ：用来通过原生 CephFS 客户端用户名进行身份验证。



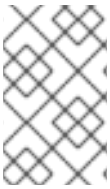
注意

访问类型取决于共享的协议。对于 CIFS，您可以使用 用户。对于 NFS 共享，您必须使用 ip。对于原生 CephFS 共享，您必须使用 cephx。

- 访问级别 ：可选；默认值为 rw。
  - rw ：对共享的读写访问。
  - ro: 对共享的只读访问权限。
- Clientidentifier: 根据 accesstype.
  - 为 ip accesstype 使用 IP 地址。
  - 使用一个 CIFS 用户或组作为 user accesstype。
  - 为 cephx accesstype 使用用户名字符串。

7.7.1. 授予对 NFS 共享的访问权限

云用户可以通过 IP 地址提供对 NFS 共享的访问。



注意

您可以将以下步骤与 IPv4 或 IPv6 地址一起使用。

流程

- 检索您计划挂载共享的客户端计算实例的 IP 地址。请确定您选择与可以访问共享的网络对应的 IP 地址。在本例中，它是 StorageNFS 网络的 IP 地址：

```
$ openstack server list -f yaml
- Flavor: m1.micro
  ID: 0b878c11-e791-434b-ab63-274ecfc957e8
  Image: manila-test
  Name: demo-instance0
  Networks: demo-network=198.51.100.4, 10.0.0.53;
  StorageNFS=198.51.100.160
```

```
  Status: ACTIVE

$ manila access-allow <share> ip 198.51.100.160
```

注意

对共享的访问权限具有自己的 ID，accessid。

```
+-----------------+--------------------------------------+
| Property        | Value                                |
+-----------------+--------------------------------------+
| access_key      | None
| share_id        | db3bedd8-bc82-4100-a65d-53ec51b5cba3
| created_at      | 2018-09-17T21:57:42.000000
| updated_at      | None
| access_type     | ip
| access_to       | 198.51.100.160
| access_level    | rw
| state           | queued_to_apply
| id              | 875c6251-c17e-4c45-8516-fe0928004fff
+-----------------+--------------------------------------+
```

验证

- 验证访问配置是否成功：

```
$ manila access-list <share>

+--------------+------------+--------------+--------------+--------+ ...
| id           | access_type | access_to    | access_level | state  | ...
+--------------+------------+--------------+--------------+--------+
| 875c6251-... | ip          | 198.51.100.160 | rw         | active | ...
+--------------+------------+--------------+--------------+---------+ ...
```

7.7.2. 授予原生 CephFS 共享的访问权限

云用户可以通过 Ceph 客户端用户名提供对原生 CephFS 共享的访问。共享文件系统服务(manila)可防止使用预先存在的 Ceph 用户，因此您必须创建唯一的 Ceph 客户端用户名。

若要挂载共享，您需要 Ceph 客户端用户名和访问密钥。您可以使用共享文件系统服务 API 检索访问密钥。默认情况下，访问密钥对项目命名空间中的所有用户可见。您可以为同一用户提供对项目命名空间中不同共享的访问权限。然后，用户可以使用客户端计算机上的 CephFS 内核客户端来访问共享。

重要

仅将原生 CephFS 驱动程序与可信客户端一起使用。有关原生 CephFS 后端安全性的信息，请参阅 *部署 Red Hat Ceph Storage 和 Red Hat OpenStack Platform* 中的 原生 CephFS 后端安全性。

流程

1. 授予用户对原生 CephFS 共享的访问权限：

```
$ manila access-allow <share> cephx <user>
```

- 将 <share > 替换为共享名称或共享 ID。
- 将 & lt;user& gt; 替换为 cephx 用户。

2. 收集用户的访问密钥：

```
$ manila access-list <share>
```

7.7.3. 授予 CIFS 共享的访问权限

云用户可以通过 Active Directory 服务中存在的用户名授予对 CIFS 共享的访问权限。共享文件系统服务(manila)不会在 Active Directory 服务器上创建新用户。它仅通过安全服务验证用户名，访问具有无效用户名的规则 会导致错误 状态。

如果云管理员将 driver_handles_share_servers 参数的值设置为 true，则云用户通过添加安全服务来配置 Active Directory 服务。如果云管理员将 DHSS 参数的值设置为 false，则云管理员将配置 Active Directory 服务并将其与存储网络相关联。

要挂载共享，您必须指定用户的 Active Directory 用户名和密码。您无法通过共享文件系统服务获取此密码。

流程

- 授予用户对 CIFS 共享的访问权限：

```
$ manila access-allow <share> user <user>
```

- 将 **<share >** 替换为共享名称或共享 **ID**。
- 将 **<user >** 替换为与 **Active Directory** 用户对应的用户名。

7.7.4. 撤销对共享的访问

共享的所有者可以撤销对共享的访问的原因。完成以下步骤以撤销之前授予共享的访问。

流程

- 撤销对共享的访问权限：

```
$ manila access-deny <share> <access-id>
```

- 将 **<share >** 替换为共享名称或共享 **ID**。
- 将 **<access-id >** 替换为共享的访问 **ID**。

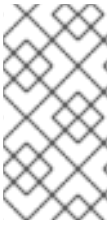
  例如：

```
$ manila access-list share-01
+--------------+------------+--------------+--------------+--------+
| id           | access_type | access_to    | access_level | state  | ...
+--------------+------------+--------------+--------------+--------+ ...
| 875c6251-... | ip          | 198.51.100.160 | rw         | active | ...
+--------------+------------+--------------+--------------+--------+

$ manila access-deny share-01 875c6251-c17e-4c45-8516-fe0928004fff

$ manila access-list share-01

+--------------+------------+--------------+--------------+--------+ ...
| id           | access_type| access_to    | access_level | state  | ...
+--------------+------------+--------------+--------------+--------+ ...
+--------------+------------+--------------+--------------+--------+ ...
```



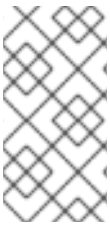
注意

如果您有一个具有读写权限的现有客户端，则必须撤销对共享的访问权限，并添加一个只读规则（如果您希望客户端具有只读权限）。

7.8. 在计算实例上挂载共享

在授予对客户端的共享访问权限后，客户端可以挂载和使用共享。任何类型的客户端都可以访问共享，只要有与客户端的网络连接。

在虚拟计算实例上挂载 NFS 共享的步骤与在裸机计算实例上挂载 NFS 共享的步骤类似。有关如何在 OpenShift 容器上挂载共享的更多信息，请参阅 OpenShift Container Platform 产品文档。



注意

用于不同协议的客户端软件包必须安装在挂载共享的 Compute 实例上。例如，对于 CephFS-NFS 的共享文件系统服务，NFS 客户端软件包必须支持 NFS 4.1。

7.8.1. 列出共享导出位置

检索共享的导出位置，以便您可以挂载共享。

流程

- 检索共享的导出位置：

```
$ manila share-export-location-list share-01
```

如果存在多个导出位置，选择一个 preferred metadata 字段的值等于 True 的位置。如果没有首选位置，您可以使用任何导出位置。

7.8.2. 挂载 NFS、原生 CephFS 或 CIFS 共享

当您创建 NFS、原生 CephFS 或 CIFS 共享并授予对最终用户客户端的共享访问权限时，用户可以在客户端上挂载共享以启用数据的访问权限，只要有网络连接。

先决条件

- 要挂载 NFS 共享，必须在客户端计算机上安装 nfs-utils 软件包。
- 若要挂载原生 CephFS 共享，必须在客户端计算机上安装 ceph-common 软件包。用户通过使用客户端计算机上的 CephFS 内核客户端来访问原生 CephFS 共享。
- 要挂载 CIFS 共享，必须在客户端计算机上安装 cifs-utils 软件包。

流程

1. 登录到实例：

```
$ openstack server ssh demo-instance0 --login user
```

2. 挂载 NFS 共享。有关示例语法，请参考以下示例：

```
$ mount -t nfs \
 -v <198.51.100.13:/volumes/_nogroup/e840b4ae-6a04-49ee-9d6e-67d4999fbc01>  \
 /mnt
```

- 将 <198.51.100.13:/volumes/_nogroup/e840b4ae-6a04-49ee-9d6e-67d4999fbc01 > 替换为共享的导出位置。
- 检索导出位置，如 第 7.8.1 节 “列出共享导出位置” 所述。

3. 挂载原生 CephFS 共享。有关示例语法，请参考以下示例：

```
$  mount -t ceph \
 <192.0.2.125:6789,192.0.2.126:6789,192.0.2.127:6789:/volumes/_nogroup/4c55ad20-9c55-
4a5e-9233-8ac64566b98c> \
 -o name=<user>,secret='<AQA8+ANW/<4ZWNRAAOtWJMFPEihBA1unFImJczA==>'
```

- 将 <192.0.2.125:6789,192.0.2.126:6789,192.0.2.127:6789:/volumes/_nogroup/4c55ad20-9c55-4a5e-9233-8ac64566b98c > 替换为共享的导出位置。

- 检索导出位置，如 第 7.8.1 节 “列出共享导出位置” 所述。
- 将 <user > 替换为有权访问共享的 cephx 用户。
- 将 secret 值替换为您在 第 7.7.2 节 “授予原生 CephFS 共享的访问权限” 中收集的访问密钥。

4. 挂载 CIFS 共享。有关示例语法，请参考以下示例：

```
$  mount -t cifs \
 -o user=<user>,pass=<password> \
 <\\192.0.2.128/share_11265e8a_200c_4e0a_a40f_b7a1117001ed>
```

- 将 <user > 替换为有权访问共享的 Active Directory 用户。
- 将 <password > 替换为用户的 Active Directory 密码。
- 将 <\\192.0.2.128/share_11265e8a_200c_4e0a_a40f_b7a1117001ed > 替换为共享的导出位置。
- 检索导出位置，如 第 7.8.1 节 “列出共享导出位置” 所述。

验证

- 验证 mount 命令是否成功：

```
$ df -k
```

7.9. 删除共享

共享文件系统服务(manila)不提供保护来防止您删除数据。共享文件系统服务不会检查客户端是否连接或工作负载是否正在运行。当您删除共享时，您无法检索它。

警告

在删除共享前备份您的数据。

先决条件

- 如果从共享创建快照，您必须先删除所有快照和副本，然后才能删除共享。如需更多信息，请参阅 删除快照。

流程

- 删除共享：

```
$ manila delete <share>
```

- 将 <share > 替换为共享名称或共享 ID。

7.10. 列出共享文件系统服务的资源限制

作为云用户，您可以列出当前的资源限值。这可帮助您规划工作负载，并根据资源消耗准备任何操作。

流程

- 列出项目的资源限值和当前资源消耗：

```
$ manila absolute-limits
+-----------------------------+-------+
| Name                        | Value |
+-----------------------------+-------+
| maxTotalReplicaGigabytes    | 1000  |
| maxTotalShareGigabytes      | 1000  |
| maxTotalShareGroupSnapshots | 50    |
| maxTotalShareGroups         | 49    |
| maxTotalShareNetworks       | 10    |
| maxTotalShareReplicas       | 100   |
| maxTotalShareSnapshots      | 50    |
| maxTotalShares              | 50    |
| maxTotalSnapshotGigabytes   | 1000  |
```

```
| totalReplicaGigabytesUsed    | 22    |
| totalShareGigabytesUsed      | 25    |
| totalShareGroupSnapshotsUsed | 0     |
| totalShareGroupsUsed         | 9     |
| totalShareNetworksUsed       | 2     |
| totalShareReplicasUsed       | 9     |
| totalShareSnapshotsUsed      | 4     |
| totalSharesUsed              | 12    |
| totalSnapshotGigabytesUsed   | 4     |
+------------------------------+-------+
```

7.11. 操作失败的故障排除

如果共享文件系统服务(manila)操作（如创建共享或创建共享组）异步失败，您可以作为最终用户从命令行运行查询，以了解更多有关错误的信息。

7.11.1. 修复创建共享或创建共享组失败

在本例中，最终用户的目标是在多个虚拟机上创建一个共享来托管软件库。示例错误地引入了两个共享创建失败，以说明如何使用命令行来检索用户支持消息。

流程

1. **要创建共享，您可以使用共享类型指定您希望共享具有的一些功能。云管理员可以创建共享类型。查看可用的共享类型：**

```
clouduser1@client:~$ manila type-list
+--------------------------------------+-------------+------------+------------+------------------------------
--------+----------------------------------------------+-------------+
| ID                                   | Name        | visibility | is_default | required_extra_specs                |
optional_extra_specs                       | Description |
+--------------------------------------+-------------+------------+------------+------------------------------
--------+----------------------------------------------+-------------+
| 1cf5d45a-61b3-44d1-8ec7-89a21f51a4d4 | dhss_false  | public     | YES        |
driver_handles_share_servers : False | create_share_from_snapshot_support : True  | None
|
|                                      |             |            |            |                                |
mount_snapshot_support : False             |             |
|                                      |             |            |            |                                |
revert_to_snapshot_support : False         |             |
|                                      |             |            |            |                                | snapshot_support :
True                  |             |
| 277c1089-127f-426e-9b12-711845991ea1 | dhss_true   | public     | -          |
driver_handles_share_servers : True  | create_share_from_snapshot_support : True  | None
|
|                                      |             |            |            |                                |
mount_snapshot_support : False             |             |
|                                      |             |            |            |                                |
```

```
revert_to_snapshot_support : False         |            |
|                                  |           |          |          |                                | snapshot_support :
True                  |           |
+----------------------------------+------------+-----------+-----------+-----------------------------
--------+-----------------------------------------+------------+
```

在本例中，有两个共享类型可用。

2.

要使用指定 driver_handles_share_servers=true 功能的共享类型，您必须创建一个共享网络来导出共享。从专用项目网络创建共享网络。

```
clouduser1@client:~$ openstack subnet list
+--------------------------------------+------------------+--------------------------------------+----------
----------+
| ID                                   | Name             | Network                              | Subnet           |
+--------------------------------------+------------------+--------------------------------------+----------
----------+
| 78c6ac57-bba7-4922-ab81-16cde31c2d06 | private-subnet     | 74d5cfb3-5dd0-43f7-b1b2-
5b544cb16212 | 10.0.0.0/26        |
| a344682c-718d-4825-a87a-3622b4d3a771 | ipv6-private-subnet | 74d5cfb3-5dd0-43f7-
b1b2-5b544cb16212 | fd36:18fc:a8e9::/64 |
+--------------------------------------+------------------+--------------------------------------+----------
----------+

clouduser1@client:~$ manila share-network-create \
  --name mynet \
  --neutron-net-id 74d5cfb3-5dd0-43f7-b1b2-5b544cb16212 \
  --neutron-subnet-id 78c6ac57-bba7-4922-ab81-16cde31c2d06
+-------------------+--------------------------------------+
| Property          | Value                                |
+-------------------+--------------------------------------+
| network_type      | None                                 |
| name              | mynet                                |
| segmentation_id   | None                                 |
| created_at        | 2018-10-09T21:32:22.485399           |
| neutron_subnet_id | 78c6ac57-bba7-4922-ab81-16cde31c2d06 |
| updated_at        | None                                 |
| mtu               | None                                 |
| gateway           | None                                 |
| neutron_net_id    | 74d5cfb3-5dd0-43f7-b1b2-5b544cb16212 |
| ip_version        | None                                 |
| cidr              | None                                 |
| project_id        | cadd7139bc3148b8973df097c0911016     |
| id                | 0b0fc320-d4b5-44a1-a1ae-800c56de550c |
| description       | None                                 |
+-------------------+--------------------------------------+

clouduser1@client:~$ manila share-network-list
+--------------------------------------+-------+
| id                                   | name  |
```

```
+--------------------------------------+-------+
| 6c7ef9ef-3591-48b6-b18a-71a03059edd5 | mynet |
+--------------------------------------+-------+
```

3. 创建共享：

```
clouduser1@client:~$ manila create nfs 1 \
  --name software_share \
  --share-network mynet \
  --share-type dhss_true
+---------------------------------------+--------------------------------------+
| Property                              | Value                                |
+---------------------------------------+--------------------------------------+
| status                                | creating                             |
| share_type_name                       | dhss_true                            |
| description                           | None                                 |
| availability_zone                     | None                                 |
| share_network_id                      | 6c7ef9ef-3591-48b6-b18a-71a03059edd5 |
| share_server_id                       | None                                 |
| share_group_id                        | None                                 |
| host                                  |                                      |
| revert_to_snapshot_support            | False                                |
| access_rules_status                   | active                               |
| snapshot_id                           | None                                 |
| create_share_from_snapshot_support    | False                                |
| is_public                             | False                                |
| task_state                            | None                                 |
| snapshot_support                      | False                                |
| id                                    | 243f3a51-0624-4bdd-950e-7ed190b53b67 |
| size                                  | 1                                    |
| source_share_group_snapshot_member_id | None                                 |
| user_id                               | 61aef4895b0b41619e67ae83fba6defe     |
| name                                  | software_share                       |
| share_type                            | 277c1089-127f-426e-9b12-711845991ea1 |
| has_replicas                          | False                                |
| replication_type                      | None                                 |
| created_at                            | 2018-10-09T21:12:21.000000           |
| share_proto                           | NFS                                  |
| mount_snapshot_support                | False                                |
| project_id                            | cadd7139bc3148b8973df097c0911016     |
| metadata                              | {}                                   |
+---------------------------------------+--------------------------------------+
```

4. 查看共享的状态：

```
clouduser1@client:~$ manila list
+--------------------------------------+---------------+------+-------------+--------+-----------+----------
-------+------+------------------+
| ID                                   | Name          | Size | Share Proto | Status | Is Public | Share Type
Name | Host | Availability Zone |
+--------------------------------------+---------------+------+-------------+--------+-----------+----------
```

```
-------+------+------------------+
| 243f3a51-0624-4bdd-950e-7ed190b53b67 | software_share | 1    | NFS         | error  | False
| dhss_true       |      | None              |
+--------------------------------------+---------------+------+-------------+--------+-----------+----------
-------+------+------------------+
```

在本例中，共享创建过程中出现错误。

5. **要查看用户支持消息，请运行 message-list 命令。使用 --resource-id 过滤到您要查找的特定共享。**

```
clouduser1@client:~$ manila message-list
+--------------------------------------+-------------+-----------------------------------+-----------+---
-------------------------------------------------------------------------------------------------+--------
---+----------------------------+
| ID                                   | Resource Type | Resource ID                       | Action ID | User
Message                                                                                          | Detail ID | Created At
|
+--------------------------------------+-------------+-----------------------------------+-----------+---
-------------------------------------------------------------------------------------------------+--------
---+----------------------------+
| 7d411c3c-46d9-433f-9e21-c04ca30b209c | SHARE         | 243f3a51-0624-4bdd-950e-
7ed190b53b67 | 001       | allocate host: No storage could be allocated for this share request,
Capabilities filter didn't succeed. | 008       | 2018-10-09T21:12:21.000000 |
+--------------------------------------+-------------+-----------------------------------+-----------+---
-------------------------------------------------------------------------------------------------+--------
---+----------------------------+
```

在 User Message 列中，请注意，由于功能不匹配，共享文件系统服务无法创建共享。

6. **要查看更多信息，请运行 message-show 命令，后跟 message-list 命令中的消息的 ID：**

```
clouduser1@client:~$ manila message-show 7d411c3c-46d9-433f-9e21-c04ca30b209c
+---------------+-----------------------------------------------------------------------------------------
-----------+
| Property      | Value                                                                                    |
+---------------+-----------------------------------------------------------------------------------------
-----------+
| request_id    | req-0a875292-6c52-458b-87d4-1f945556feac
|
| detail_id     | 008                                                                                      |
| expires_at    | 2018-11-08T21:12:21.000000
|
| resource_id   | 243f3a51-0624-4bdd-950e-7ed190b53b67
|
| user_message  | allocate host: No storage could be allocated for this share request,
Capabilities filter didn't succeed. |
| created_at    | 2018-10-09T21:12:21.000000
```

```
|
| message_level | ERROR                                                                                                |
| id            | 7d411c3c-46d9-433f-9e21-c04ca30b209c
|
| resource_type | SHARE                                                                                                |
| action_id     | 001                                                                                                  |
+---------------+------------------------------------------------------------------------------------------------------
----------+
```

7. **作为云用户，您可以通过共享类型检查功能，以便您可以查看可用的共享类型。两种共享类型之间的区别是 driver_handles_share_servers 的值：**

```
clouduser1@client:~$ manila type-list
+--------------------------------------+------------+------------+------------+-------------------------------
--------+----------------------------------------------+------------+
| ID                                   | Name       | visibility | is_default | required_extra_specs                |
optional_extra_specs                         | Description |
+--------------------------------------+------------+------------+------------+-------------------------------
--------+----------------------------------------------+------------+
| 1cf5d45a-61b3-44d1-8ec7-89a21f51a4d4 | dhss_false | public     | YES        |
driver_handles_share_servers : False | create_share_from_snapshot_support : True  | None
|
|                                      |            |            |            |                                     |
mount_snapshot_support : False             |            |
|                                      |            |            |            |                                     |
revert_to_snapshot_support : False         |            |
|                                      |            |            |            |                                     | snapshot_support :
True                  |            |
| 277c1089-127f-426e-9b12-711845991ea1 | dhss_true  | public     | -          |
driver_handles_share_servers : True  | create_share_from_snapshot_support : True  | None
|
|                                      |            |            |            |                                     |
mount_snapshot_support : False             |            |
|                                      |            |            |            |                                     |
revert_to_snapshot_support : False         |            |
|                                      |            |            |            |                                     | snapshot_support :
True                  |            |
+--------------------------------------+------------+------------+------------+-------------------------------
--------+----------------------------------------------+------------+
```

8. **使用其他可用共享类型创建共享：**

```
clouduser1@client:~$ manila create nfs 1 \
  --name software_share \
  --share-network mynet \
  --share-type dhss_false
+--------------------------------------+--------------------------------------+
| Property                             | Value                                |
+--------------------------------------+--------------------------------------+
| status                               | creating                             |
| share_type_name                      | dhss_false                           |
```

```
| description                         | None                                 |
| availability_zone                   | None                                 |
| share_network_id                    | 6c7ef9ef-3591-48b6-b18a-71a03059edd5 |
| share_group_id                      | None                                 |
| revert_to_snapshot_support          | False                                |
| access_rules_status                 | active                               |
| snapshot_id                         | None                                 |
| create_share_from_snapshot_support  | True                                 |
| is_public                           | False                                |
| task_state                          | None                                 |
| snapshot_support                    | True                                 |
| id                                  | 2d03d480-7cba-4122-ac9d-edc59c8df698 |
| size                                | 1                                    |
| source_share_group_snapshot_member_id | None                                 |
| user_id                             | 5c7bdb6eb0504d54a619acf8375c08ce     |
| name                                | software_share                       |
| share_type                          | 1cf5d45a-61b3-44d1-8ec7-89a21f51a4d4 |
| has_replicas                        | False                                |
| replication_type                    | None                                 |
| created_at                          | 2018-10-09T21:24:40.000000           |
| share_proto                         | NFS                                  |
| mount_snapshot_support              | False                                |
| project_id                          | cadd7139bc3148b8973df097c0911016     |
| metadata                            | {}                                   |
+-------------------------------------+--------------------------------------+
```

在本例中，第二个共享创建尝试失败。

9. **查看用户支持消息：**

```
clouduser1@client:~$ manila list
+--------------------------------------+----------------+------+-------------+--------+-----------+----------
-------+------+------------------+
| ID                                   | Name           | Size | Share Proto | Status | Is Public | Share Type
Name | Host | Availability Zone |
+--------------------------------------+----------------+------+-------------+--------+-----------+----------
-------+------+------------------+
| 2d03d480-7cba-4122-ac9d-edc59c8df698 | software_share | 1    | NFS         | error  | False
| dhss_false      |      | nova              |
| 243f3a51-0624-4bdd-950e-7ed190b53b67 | software_share | 1    | NFS         | error  | False
| dhss_true       |      | None              |
+--------------------------------------+----------------+------+-------------+--------+-----------+----------
-------+------+------------------+

clouduser1@client:~$ manila message-list
+--------------------------------------+--------------+-------------------------------------+-----------+---
-----------------------------------------------------------------------------------------------+--------
---+----------------------------+
| ID                                   | Resource Type | Resource ID                         | Action ID | User
Message                                                                                        | Detail ID | Created At
|
+--------------------------------------+--------------+-------------------------------------+-----------+---
```

```
---------------------------------------------------------------------------------------------------+--------
---+---------------------------+
| ed7e02a2-0cdb-4ff9-b64f-e4d2ec1ef069 | SHARE         | 2d03d480-7cba-4122-ac9d-
edc59c8df698 | 002       | create: Driver does not expect share-network to be provided with
current configuration.                  | 003       | 2018-10-09T21:24:40.000000 |
| 7d411c3c-46d9-433f-9e21-c04ca30b209c | SHARE         | 243f3a51-0624-4bdd-950e-
7ed190b53b67 | 001       | allocate host: No storage could be allocated for this share request,
Capabilities filter didn't succeed. | 008       | 2018-10-09T21:12:21.000000 |
+--------------------------------------+--------------+-----------------------------------+-----------+---
---------------------------------------------------------------------------------------------------+--------
---+---------------------------+
```

该服务不会预期您所使用的共享类型的共享网络。

10.

如果没有咨询管理员，您可以发现管理员尚未提供支持直接将共享导出到私有 neutron 网络的存储后端。创建没有 share-network 参数的共享：

```
clouduser1@client:~$ manila create nfs 1 \
  --name software_share \
  --share-type dhss_false
+-------------------------------------+-----------------------------------+
| Property                            | Value                             |
+-------------------------------------+-----------------------------------+
| status                              | creating                          |
| share_type_name                     | dhss_false                        |
| description                         | None                              |
| availability_zone                   | None                              |
| share_network_id                    | None                              |
| share_group_id                      | None                              |
| revert_to_snapshot_support          | False                             |
| access_rules_status                 | active                            |
| snapshot_id                         | None                              |
| create_share_from_snapshot_support  | True                              |
| is_public                           | False                             |
| task_state                          | None                              |
| snapshot_support                    | True                              |
| id                                  | 4d3d7fcf-5fb7-4209-90eb-9e064659f46d |
| size                                | 1                                 |
| source_share_group_snapshot_member_id | None                            |
| user_id                             | 5c7bdb6eb0504d54a619acf8375c08ce  |
| name                                | software_share                    |
| share_type                          | 1cf5d45a-61b3-44d1-8ec7-89a21f51a4d4 |
| has_replicas                        | False                             |
| replication_type                    | None                              |
| created_at                          | 2018-10-09T21:25:40.000000        |
| share_proto                         | NFS                               |
| mount_snapshot_support              | False                             |
| project_id                          | cadd7139bc3148b8973df097c0911016  |
| metadata                            | {}                                |
+-------------------------------------+-----------------------------------+
```

11. **确保共享创建成功：**

```
clouduser1@client:~$ manila list
+--------------------------------------+---------------+------+-------------+-----------+-----------+-------
----------+------+-------------------+
| ID                                   | Name          | Size | Share Proto | Status    | Is Public | Share Type
Name | Host | Availability Zone |
+--------------------------------------+---------------+------+-------------+-----------+-----------+-------
----------+------+-------------------+
| 4d3d7fcf-5fb7-4209-90eb-9e064659f46d | software_share | 1    | NFS         | available |
False     | dhss_false      |      | nova              |
| 2d03d480-7cba-4122-ac9d-edc59c8df698 | software_share | 1    | NFS         | error     | False
| dhss_false      |      | nova              |
| 243f3a51-0624-4bdd-950e-7ed190b53b67 | software_share | 1    | NFS         | error     |
False     | dhss_true       |      | None              |
+--------------------------------------+---------------+------+-------------+-----------+-----------+-------
----------+------+-------------------+
```

12. **删除共享和支持信息：**

```
clouduser1@client:~$ manila message-list
+--------------------------------------+--------------+--------------------------------------+-----------+---
-------------------------------------------------------------------------------------------------+--------
---+----------------------------+
| ID                                   | Resource Type | Resource ID                          | Action ID | User
Message                                                                                          | Detail ID | Created At
|
+--------------------------------------+--------------+--------------------------------------+-----------+---
-------------------------------------------------------------------------------------------------+--------
---+----------------------------+
| ed7e02a2-0cdb-4ff9-b64f-e4d2ec1ef069 | SHARE         | 2d03d480-7cba-4122-ac9d-
edc59c8df698 | 002       | create: Driver does not expect share-network to be provided with
current configuration.                  | 003       | 2018-10-09T21:24:40.000000 |
| 7d411c3c-46d9-433f-9e21-c04ca30b209c | SHARE         | 243f3a51-0624-4bdd-950e-
7ed190b53b67 | 001       | allocate host: No storage could be allocated for this share request,
Capabilities filter didn't succeed. | 008       | 2018-10-09T21:12:21.000000 |
+--------------------------------------+--------------+--------------------------------------+-----------+---
-------------------------------------------------------------------------------------------------+--------
---+----------------------------+

clouduser1@client:~$ manila delete 2d03d480-7cba-4122-ac9d-edc59c8df698 243f3a51-
0624-4bdd-950e-7ed190b53b67
clouduser1@client:~$ manila message-delete ed7e02a2-0cdb-4ff9-b64f-e4d2ec1ef069
7d411c3c-46d9-433f-9e21-c04ca30b209c

clouduser1@client:~$ manila message-list
+----+---------------+-------------+-----------+--------------+-----------+------------+
| ID | Resource Type | Resource ID | Action ID | User Message | Detail ID | Created At |
+----+---------------+-------------+-----------+--------------+-----------+------------+
+----+---------------+-------------+-----------+--------------+-----------+------------+
```

7.11.2. 调试共享挂载失败

如果您在挂载共享时遇到问题，请使用这些验证步骤来识别根本原因。

流程

1. 验证共享的访问控制列表，以确保与您的客户端对应的规则正确且已成功应用。

```
$ manila access-list share-01
```

在成功规则中，state 属性等于 active。

2. 如果 share type 参数被配置为 driver_handles_share_servers=false，请从导出位置复制主机名或 IP 地址，并 ping 以确认与 NAS 服务器的连接：

```
$ ping -c 1 198.51.100.13
PING 198.51.100.13 (198.51.100.13) 56(84) bytes of data.
64 bytes from 198.51.100.13: icmp_seq=1 ttl=64 time=0.048 ms--- 198.51.100.13 ping
statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 7.851/7.851/7.851/0.000 ms
If using the NFS protocol, you may verify that the NFS server is ready to respond to NFS rpcs
on the proper port:
$ rpcinfo -T tcp -a 198.51.100.13.8.1 100003 4
program 100003 version 4 ready and waiting
```

注意

IP 地址以通用地址格式(uaddr)编写，它添加了两个额外的 octets (8.1)来代表 NFS 服务端口 2049。

如果这些验证步骤失败，则可能存在网络连接问题，或者您的共享文件系统后端存储失败。收集日志文件并联系红帽支持。