Red Hat Ceph Storage 4

安装指南

在 Red Hat Enterprise Linux 上安装 Red Hat Ceph Storage

Last Updated: 2024-01-31

Red Hat Ceph Storage 4 安装指南

在 Red Hat Enterprise Linux 上安装 Red Hat Ceph Storage

法律通告

摘要

本文档提供在 AMD64 和 Intel 64 架构上运行的 Red Hat Enterprise Linux 7 和 Red Hat Enterprise Linux 8 安装 Red Hat Ceph Storage 的说明。 红帽承诺替换我们的代码、文档和网页属性中存在问题的语言。我们从这四个术语开始： master、slave、blacklist 和 whitelist。这些更改将在即将发行的几个发行本中逐渐实施。详情请查看 CTO Chris Wright 信息。

目录

第 1 章 什么是 RED HAT CEPH STORAGE？

Red Hat Ceph Storage 是一个可扩展、开放、软件定义型存储平台，它将 Ceph 存储系统的企业级强化版本与 Ceph 管理平台、部署实用程序和支持服务相结合。Red Hat Ceph Storage 存储专为云基础架构和 Web 规模对象存储而设计。Red Hat Ceph Storage 集群由以下类型的节点组成：

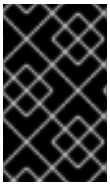
Red Hat Ceph Storage Ansible 管理

Ansible 管理节点取代了之前版本的 Red Hat Ceph Storage 中使用的传统 Ceph 管理节点。Ansible 管理节点提供以下功能：

- 集中存储集群管理。
- Ceph 配置文件和密钥。
- （可选）用于在因安全原因无法访问互联网的节点上安装 Ceph 的本地存储库。

Ceph monitor

每一 Ceph 监控 (Monitor) 节点会运行 **ceph-mon** 守护进程，它会维护存储集群映射的一个主（master）副本。存储集群映射包含存储集群拓扑。连接 Ceph 存储集群的客户端从 Ceph monitor 检索存储集群映射的当前副本，这使得客户端能够从存储集群读取和写入数据。



重要

存储群集只能使用一个 Ceph monitor 运行；但是，为了确保在生产存储集群中实现高可用性，红帽将仅支持具有至少三个 Ceph 监控节点的部署。红帽建议为超过 750 个 Ceph OSD 的存储集群部署总计 5 个 Ceph 监控器。

Ceph OSD

每个 Ceph 对象存储设备 (OSD) 节点运行 **ceph-osd** 守护进程，该守护进程与附加到节点的逻辑卷交互。存储集群在这些 Ceph OSD 节点上存储数据。

Ceph 可在只有很少 OSD 节点的环境中运行，默认为三个。但对于生产环境，只有从中等范围环境开始才可能看到其在性能方面的优势。例如，存储集群中的 50 个 Ceph OSD。理想情况下，Ceph 存储集群具有多个 OSD 节点，可以通过相应地配置 CRUSH map 来隔离故障域。

Ceph MDS

每个 Ceph 元数据服务器 (MDS) 节点运行 **ceph-mds** 守护进程，它管理与 Ceph 文件系统 (CephFS) 中存储的文件相关的元数据。Ceph MDS 守护进程也协调对共享存储集群的访问。

Ceph 对象网关

Ceph 对象网关节点运行 **ceph-radosgw** 守护进程，它是基于 **librados** 构建的对象存储接口，为应用提供 Ceph 存储集群的 RESTful 访问点。Ceph 对象网关支持两个接口：

- S3
  通过与 Amazon S3 RESTful API 的大子集兼容的接口提供对象存储功能。
- Swift
  通过与 OpenStack Swift API 的大集兼容的接口提供对象存储功能。

其它资源

- 有关 Ceph 架构的详细信息，请参阅 *Red Hat Ceph Storage 架构指南*。

- 有关最低硬件建议，请参阅 *Red Hat Ceph Storage Hardware Selection Guide*。

第 2 章 RED HAT CEPH STORAGE 注意事项和建议

作为存储管理员，您可以在运行 Red Hat Ceph Storage集群对其有一定的了解。了解诸如硬件和网络要求等因素，了解哪种类型的工作负载与 Red Hat Ceph Storage 集群配合工作以及红帽的建议。Red Hat Ceph Storage 可根据特定业务需求或一组要求，用于不同的工作负载。在安装 Red Hat Ceph Storage 之前，进行必要的规划是高效运行 Ceph 存储集群以满足业务需求的关键。

注意

您是否想要获得针对特定用例规划 Red Hat Ceph Storage 集群的帮助？请联系您的红帽代表以获得帮助。

2.1. 先决条件

- 了解、考虑和规划存储解决方案的时间.

2.2. RED HAT CEPH STORAGE 的基本注意事项

使用 Red Hat Ceph Storage 的第一个考虑因素是为数据制定存储策略。存储策略是一种存储服务特定用例的数据的方法。如果您需要为 OpenStack 等云平台存储卷和镜像，可以选择将数据存储在带有 Solid State Drives (SSD) 的快速 Serial Attached SCSI (SAS) 驱动器上。相反，如果您需要存储 S3 或 Swift 兼容网关的对象数据，您可以选择使用更经济的方式，如传统的 SATA 驱动器。Red Hat Ceph Storage 可以在同一存储集群中同时容纳这两种场景，但您需要一种方式为云平台提供快速存储策略，并为对象存储提供更传统的存储方式。

一个成功的 Ceph 部署中的最重要的一个步骤是，找出一个适合存储集群的用例和工作负载的性价比配置集。为用例选择正确的硬件非常重要。例如，为冷存储应用程序选择 IOPS 优化的硬件会不必要地增加硬件成本。然而，在 IOPS 密集型工作负载中，选择容量优化的硬件使其更具吸引力的价格点可能会导致用户对性能较慢的抱怨。

Red Hat Ceph Storage 可以支持多种存储策略。用例、成本与好处性能权衡以及数据持久性是帮助开发合理存储策略的主要考虑因素。

使用案例

Ceph 提供大量存储容量，它支持许多用例，例如：

- Ceph 块设备客户端是云平台的领先存储后端，可为具有写时复制（copy-on-write）克隆等高性能功能的卷和镜像提供无限存储。
- Ceph 对象网关客户端是云平台的领先存储后端，为音频、位映射、视频和其他数据等对象提供 RESTful S3 兼容和 Swift 兼容对象存储。
- 传统文件存储的 Ceph 文件系统.

成本比较性能优势

越快越好。越大越好。越耐用越好。但是，每种出色的质量、相应的成本与收益权衡都有价格。从性能角度考虑以下用例：SSD 可以为相对较小的数据和日志量提供非常快速的存储。存储数据库或对象索引可以从非常快的 SSD 池中受益，但对于其他数据而言成本过高。带有 SSD 日志的 SAS 驱动器以经济的价格为卷和图像提供快速性能。没有 SSD 日志地 SATA 驱动器可提供低成本存储，同时整体性能也较低。在创建 OSD 的 CRUSH 层次结构时，您需要考虑用例和可接受的成本与性能权衡。

数据持续时间

在大型存储集群中，硬件故障是预期的，而非例外。但是，数据丢失和服务中断仍然不可接受。因此，数

据的持久性非常重要。Ceph 通过对象的多个副本解决数据持久性问题，或使用纠删代码和多个编码区块来解决数据持久性。多个副本或多个编码区块会带来额外的成本与好处权衡：存储更少的副本或编码区块会更便宜，但可能会导致在降级状态中为写入请求提供服务。通常，一个具有两个额外副本的对象（或两个编码区块）可以允许存储集群在存储集群恢复时服务降级状态的写入。

在出现硬件故障时，复制存储在故障域中的一个或多个数据冗余副本。但是，冗余的数据副本规模可能会变得昂贵。例如，要存储 1 PB 字节并带有三倍复制的数据，将需要至少具有 3 PB 存储容量的集群。

纠删代码将数据存储为数据区块和编码区块。如果数据区块丢失，纠删代码可以使用剩余的数据区块和编码区块来恢复丢失的数据区块。纠删代码比复制更经济。例如，使用带有 8 个数据区块和 3 个编码区块的纠删代码提供与 3 个数据副本相同的冗余。但是，与复制相比（使用 3 倍的初始数据），此类编码方案使用约 1.5 倍的初始数据。

CRUSH 算法通过确保 Ceph 将额外的副本或编码区块存储在存储集群内的不同位置来协助这个过程。这样可确保单个存储设备或节点的故障不会丢失防止数据丢失所需的所有副本或编码区块。您可以规划一个成本取舍存储策略，以及数据持久性，然后将它作为存储池呈现给 Ceph 客户端。

重要

数据存储池可以使用纠删代码。存储服务数据和存储桶索引的池使用复制。

重要

与 Ceph 的对象复制或编码区块相比，RAID 解决方案已变得过时。不要使用 RAID，因为 Ceph 已经处理数据持久性，降级的 RAID 对性能有负面影响，并且使用 RAID 恢复数据比使用深度副本或纠删代码区块要慢得多。

其它资源

- 如需了解更多详细信息，请参阅 *《红帽 Ceph 存储 安装指南》中有关红帽 Ceph 存储 的硬件注意事项* 部分。

2.3. RED HAT CEPH STORAGE 工作负载注意事项

Ceph 存储集群的一个关键优势在于能够使用性能域支持同一存储集群中的不同类型的工作负载。不同的硬件配置可以与每个性能域关联。存储管理员可以在适当的性能域中部署存储池，为应用提供专为特定性能和成本配置文件量身定制的存储。为这些性能域选择适当的大小和优化的服务器是设计 Red Hat Ceph Storage 集群的一个重要方面。

在读取和写入数据的 Ceph 客户端接口中，Ceph 存储集群显示为一个客户端存储数据的简单池。但是，存储集群以对客户端接口完全透明的方式执行许多复杂的操作。Ceph 客户端和 Ceph 对象存储守护进程（称为 Ceph OSD）或只是 OSD，都使用可扩展哈希下的受控复制 (CRUSH) 算法来存储和检索对象。Ceph OSD 可以使用容器或基于 RPM 的部署在存储集群内的裸机服务器或虚拟机上运行。

CRUSH map 描述了集群资源的拓扑结构，并且 map 存在于客户端节点和集群中的 Ceph 监控节点中。Ceph 客户端和 Ceph OSD 都使用 CRUSH map 和 CRUSH 算法。Ceph 客户端直接与 OSD 通信，消除了集中式对象查找和潜在的性能瓶颈。利用 CRUSH map 并与其对等方通信，OSD 可以处理复制、回填和恢复，从而实现动态故障恢复。

Ceph 使用 CRUSH map 来实施故障域。Ceph 还使用 CRUSH map 实施性能域，这只需将底层硬件的性能配置文件纳入考量。CRUSH map 描述了 Ceph 存储数据的方式，它作为简单的层次结构（特别是圆环图和规则集）实施。CRUSH map 可以支持多种层次结构，将一种类型的硬件性能配置集与另一类分隔开。Ceph 实施具有设备"类"的性能域。

例如，您可以让这些性能域共存在同一 Red Hat Ceph Storage 集群中：

- 硬盘 (HDD) 通常适合以成本和容量为导向的工作负载。
- 吞吐量敏感的工作负载通常使用 HDD，在固态驱动器 (SSD) 上 Ceph 写入日志。
- MySQL 和 MariaDB 等 IOPS 密集型工作负载通常使用 SSD。

工作负载

Red Hat Ceph Storage 针对三种主要工作负载进行了优化：

- **优化 IOPS：** IOPS（Input, output per second）优化部署适合云计算操作，例如将 MYSQL 或 MariaDB 实例作为 OpenStack 上的虚拟机运行。优化 IOPS 部署需要更高的性能存储，如 15k RPM SAS 驱动器和单独的 SSD 日志，以处理频繁的写入操作。一些高 IOPS 情景使用所有闪存存储来提高 IOPS 和总吞吐量。
  IOPS 优化存储集群具有以下属性：
  - 每个 IOPS 的成本最低。
  - 每 GB 的 IOPS 最高。
  - 99 个百分点延迟一致性.

  IOPS 优化存储集群的用例：
  - 典型的块存储.
  - 用于硬盘 (HDD) 或 2x 复制的 3 倍复制，用于固态硬盘 (SSD)。
  - OpenStack 云上的 MySQL.
- **优化吞吐量：** 使用优化吞吐量的部署适合服务大量数据，如图形、音频和视频内容。优化吞吐量的部署需要高带宽网络硬件、控制器和硬盘，具有快速顺序的读写特征。如果要求快速数据访问，则使用吞吐量优化存储策略。此外，如果要求快速写入性能，将 Solid State Disk (SSD) 用于日志将显著提高写入性能。
  吞吐量优化存储集群具有以下属性：
  - 每 MBps 成本最低（吞吐量）。
  - 每个 TB 的 MBps 最高。
  - 每个 BTU 的 MBps 最高。
  - 每个 Watt 的 MBps 最高。
  - 97% 的延迟一致性.

  优化吞吐量的存储集群用例：
  - 块或对象存储。
  - 3 倍复制。
  - 面向视频、音频和图像的主动性能存储.
  - 流媒体，如 4k 视频.

- **优化容量：** 容量优化部署适合以尽可能低的成本存储大量数据。容量优化的部署通常会以更具吸引力的价格点来换取性能。例如，容量优化部署通常使用速度较慢且成本更低的 SATA 驱动器和共同定位日志，而不是使用 SSD 进行日志。
  成本和容量优化存储集群具有以下属性：

  - 每 TB 成本最低。
  - 每 TB 的 BTU 最低。
  - 每 TB 的 Watts 最低.

  成本和容量优化存储集群的用例：

  - 典型的对象存储.
  - 纠删代码，以最大程度地提高可用容量
  - 对象存档。
  - 视频、音频和图像对象存储库.

重要

在购买硬件前，请仔细考虑由 Red Hat Ceph Storage 运行的工作负载，因为它可能会显著影响存储集群的价格和性能。例如，如果工作负载是容量优化的，并且硬件更适合通过吞吐量优化的工作负载，则硬件的成本将超过必要成本。相反，如果工作负载被优化吞吐量，且硬件更适合容量优化的工作负载，则存储集群的性能会受到影响。

2.4. RED HAT CEPH STORAGE 的网络注意事项

云存储解决方案的一个重要方面是存储集群可能会因为网络延迟及其他因素而耗尽 IOPS。另外，存储集群可能会因为带宽限制而无法在存储集群用尽存储容量前耗尽吞吐量。这意味着网络硬件配置必须支持所选工作负载，以满足价格与性能要求。

存储管理员希望存储集群尽快恢复。仔细考虑存储集群网络的带宽要求、通过订阅的网络链接，以及隔离客户端到集群流量的集群内部流量。在考虑使用 Solid State Disks(SSD)、闪存、NVMe 和其他高性能存储设备时，还需要考虑到网络性能变得越来越重要。

Ceph 支持公共网络和存储集群网络。公共网络处理客户端流量以及与 Ceph 监控器的通信。存储集群网络处理 Ceph OSD 心跳、复制、回填和恢复流量。**至少**，存储硬件应使用 10 GB 的以太网链接，您可以为连接和吞吐量添加额外的 10 GB 以太网链接。

重要

红帽建议为存储集群网络分配带宽，因此它是将 **osd_pool_default_size** 用作复制池的多个基础的公共网络的倍数。红帽还建议在单独的网卡中运行公共和存储集群网络。

重要

红帽建议在生产环境中使用 10 GB 以太网部署 Red Hat Ceph Storage。1 GB 以太网网络不适用于生产环境的存储集群。

如果出现驱动器故障，在 1 GB 以太网网络中复制 1 TB 数据需要 3 小时，3 TB 需要 9 小时。使用 3 TB 是典型的驱动器配置。相比之下，使用 10 GB 以太网网络，复制时间分别为 20 分钟和 1 小时。请记住，当 Ceph OSD 出现故障时，存储集群将通过将其包含的数据复制到池中的其他 Ceph OSD 来进行恢复。

对于大型环境（如机架）的故障，意味着存储集群将使用的带宽要高得多。在构建由多个机架组成的存储群集（对于大型存储实施常见）时，应考虑在"树树"设计中的交换机之间利用尽可能多的网络带宽，以获得最佳性能。典型的 10 GB 以太网交换机有 48 10 GB 端口和四个 40 GB 端口。使用 40 GB 端口以获得最大吞吐量。或者，考虑将未使用的 10 GB 端口和 QSFP+ 和 SFP+ 电缆聚合到 40 GB 端口，以连接到其他机架和机械路由器。此外，还要考虑使用 LACP 模式 4 来绑定网络接口。另外，使用巨型帧、最大传输单元 (MTU) 9000，特别是在后端或集群网络上。

在安装和测试 Red Hat Ceph Storage 集群之前，请验证网络吞吐量。Ceph 中大多数与性能相关的问题通常是因为网络问题造成的。简单的网络问题（如粒度或 Bean Cat-6 电缆）可能会导致带宽下降。至少将 10 GB 以太网用于前端网络。对于大型集群，请考虑将 40 GB ethernet 用于后端或集群网络。

重要

为了优化网络，红帽建议使用巨型帧来获得更高的每带宽比率的 CPU，以及一个非阻塞的网络交换机后端。Red Hat Ceph Storage 在通信路径的所有网络设备中，公共和集群网络需要相同的 MTU 值。在在生产环境中使用 Red Hat Ceph Storage 集群之前，验证环境中所有节点和网络设备上的 MTU 值相同。

其它资源

- 如需了解更多详细信息，请参阅 *红帽 Ceph 存储配置指南* 中的 *验证和配置 MTU 值* 部分。

2.5. 在运行 CEPH 时调整 LINUX 内核的注意事项

生产环境的 Red Hat Ceph Storage 集群通常受益于操作系统调优，尤其是关于限值和内存分配。确保为存储集群内的所有节点设置了调整。您还可以在红帽支持下创建一个问题单，寻求其他指导。

为 Ceph OSD 保留可用内存

为了帮助防止 Ceph OSD 内存分配请求期间与内存相关的错误不足，请设置特定数量的物理内存来保留。红帽建议根据系统 RAM 数量进行以下设置。

- 对于 64 GB，保留 1 GB：

```
vm.min_free_kbytes = 1048576
```

- 对于 128 GB，保留 2 GB：

```
vm.min_free_kbytes = 2097152
```

- 对于 256 GB，保留 3 GB：

```
vm.min_free_kbytes = 3145728
```

增加文件描述符数量

如果 Ceph 对象网关缺少文件描述符，它可能会挂起。您可以修改 Ceph 对象网关节点上的 **/etc/security/limits.conf** 文件，以增加 Ceph 对象网关的文件描述符。

```
ceph      soft    nofile     unlimited
```

调整大型存储集群的 ulimit 值

在大型存储集群上运行 Ceph 管理命令时，例如，带有 1024 个 Ceph OSD 或更多 OSD，在每个运行管理命令的节点上创建一个 **/etc/security/limits.d/50-ceph.conf** 文件，其中包含以下内容：

```
USER_NAME    soft    nproc    unlimited
```

将 *USER_NAME* 替换为运行 Ceph 管理命令的非 root 用户帐户的名称。

注意

在 Red Hat Enterprise Linux 中，root 用户的 **ulimit** 值默认设置为 **ulimit**。

2.6. 将 RAID 控制器用于 OSD 节点的注意事项

另外，您也可以考虑在 OSD 节点上使用 RAID 控制器。以下是需要考虑的一些事项：

- 如果 OSD 节点安装了 1-2GB 缓存，启用回写缓存可能会导致小 I/O 写入吞吐量增加。但是，缓存必须具有非易失性。
- 大多数现代 RAID 控制器都具有超大容量，在出现电源不足时有足够的能力为非易失性 NAND 内存排空易失性内存。务必要了解特定控制器及其固件在恢复电源后的行为。
- 有些 RAID 控制器需要手动干预。硬盘驱动器通常会向操作系统播发其磁盘缓存，无论是默认应启用或禁用其磁盘缓存。但是，某些 RAID 控制器和某些固件不提供此类信息。验证磁盘级别的缓存是否已禁用，以避免文件系统损坏。
- 为启用了回写缓存的每个 Ceph OSD 数据驱动器创建一个 RAID 0 卷。
- 如果 RAID 控制器中也存在 Serial Attached SCSI(SAS)或 SATA 连接的 Solid-state Drive(SSD)磁盘，然后调查控制器和固件是否支持*透传（pass-through）*模式。启用*透传*模式有助于避免缓存逻辑，通常会降低快速介质的延迟。

2.7. 在对象网关中使用 NVME 的注意事项

另外，您可以选择将 NVMe 用于 Ceph 对象网关。

如果您计划使用 Red Hat Ceph Storage 的对象网关功能，并且 OSD 节点使用基于 NVMe 的 SSD，请按照 *Ceph Object Gateway for Production Guide* 中的 *Using NVMe with LVM optimally* 部分进行操作。这些步骤解释了如何使用专门设计的 Ansible playbook 将日志和 bucket 索引放在 SSD 上，这可以提高性能，与将所有日志放在一个设备上相比。

2.8. RED HAT CEPH STORAGE 的最低硬件注意事项

Red Hat Ceph Storage 可在非专有商用硬件上运行。通过使用适度的硬件，可在不优化性能的情况下运行小型生产集群和开发集群。

根据裸机或容器化部署，Red Hat Ceph Storage 的要求略有不同。

注意

磁盘空间要求基于 **/var/lib/ceph/** 目录下的 Ceph 守护进程默认路径。

表 2.1. 裸机

Process	标准	最低建议
ceph-osd	处理器	1x AMD64 或 Intel 64
	RAM	对于 **BlueStore** OSD，红帽通常建议每个 OSD 主机具有 16 GB RAM，每个守护进程具有额外的 5 GB RAM。
	OS Disk	每个主机 1 个 OS 磁盘
	卷存储	每个守护进程 1x 存储驱动器
	block.db	可选，但红帽建议每个守护进程 1x SSD 或 NVMe 或 Optane 分区或逻辑卷。大小为用于对象、文件和混合工作负载的 **block.data** 以及用于 **块设备、Openstack cinder 和 Openstack cinder 工作负载的 BlueStore 的 block.data** 的大小。
	block.wal	可选，每个守护进程 1x SSD 或 NVMe 或 Optane 分区或逻辑卷。只有在速度比 **block.db** 设备快时，才使用一个小的大小值（如 10 GB）。
	网络	2x 10 GB 以太网 NIC
ceph-mon	处理器	1x AMD64 或 Intel 64
	RAM	每个守护进程 1 GB
	磁盘空间	每个守护进程 15 GB
	监控磁盘	（可选）1x SSD 磁盘用于 **leveldb** 监控数据。
	网络	2x 1 GB 以太网 NIC
ceph-mgr	处理器	1x AMD64 或 Intel 64
	RAM	每个守护进程 1 GB
	网络	2x 1 GB 以太网 NIC
ceph-radosgw	处理器	1x AMD64 或 Intel 64
	RAM	每个守护进程 1 GB
	磁盘空间	每个守护进程 5 GB
	网络	1 个 1 GB 以太网 NIC
ceph-mds	处理器	1x AMD64 或 Intel 64

Process	标准	最低建议
	RAM	每个守护进程 2 GB 这个数字高度依赖于可配置的 MDS 缓存大小。RAM 要求通常为 **mds_cache_memory_limit** 配置设置中设置的两倍。另请注意，这是守护进程的内存，而不是整体系统内存。
	磁盘空间	每个守护进程 2 MB，以及日志记录所需的任何空间，这些空间可能因配置的日志级别而异。
	网络	2x 1 GB 以太网 NIC 注意这与 OSD 的网络相同。如果您在 OSD 上有一个 10 GB 网络，则应在 MDS 上使用相同的网络，这样 MDS 在涉及延迟时不会产生判断。

表 2.2. 容器

Process	标准	最低建议
ceph-osd-container	处理器	每个 OSD 容器 1 个 AMD64 或 Intel 64 CPU CORE
	RAM	每个 OSD 容器最少 5 GB RAM
	OS Disk	每个主机 1 个 OS 磁盘
	OSD 存储	每个 OSD 容器 1X 存储驱动器。无法与 OS 磁盘共享。
	block.db	可选，但红帽建议每个守护进程 1x SSD 或 NVMe 或 Optane 分区或 lvm。大小为用于对象、文件和混合工作负载的 **block.data** 以及用于 **块设备、Openstack cinder 和 Openstack cinder 工作负载的 BlueStore 的 block.data** 的大小。
	block.wal	（可选）每个守护进程 1 个 SSD 或 NVMe 或 Optane 分区或逻辑卷。只有在速度比 **block.db** 设备快时，才使用一个小的大小值（如 10 GB）。
	网络	2 个 10 GB 以太网 NIC，建议 10 GB
ceph-mon-container	处理器	每个 mon-container 1 个 AMD64 或 Intel 64 CPU 内核
	RAM	每个 **mon-container** 3 GB
	磁盘空间	每个 **mon-container** 10 GB，但推荐 50 GB
	监控磁盘	另外，1 个 SSD 磁盘用于 **monitor rocksdb** 数据

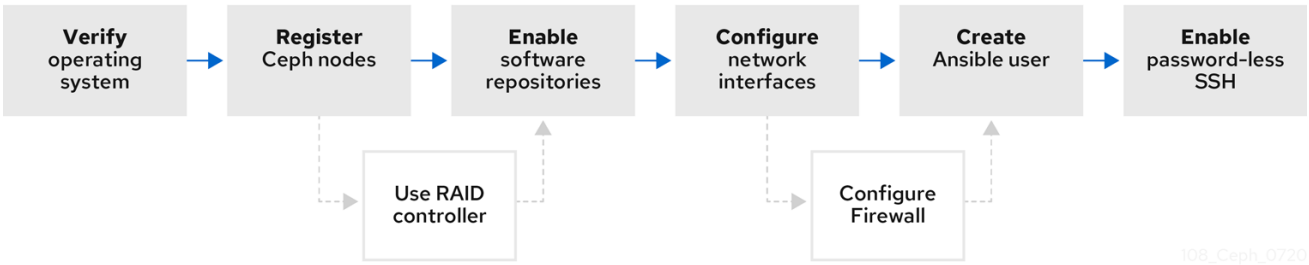
Process	标准	最低建议
	网络	2 个 1 GB 以太网 NIC，建议 10 GB
ceph-mgr-container	处理器	每个 **mgr-container** 1 个 AMD64 或 Intel 64 CPU 内核
	RAM	每个 **mgr-container** 3 GB
	网络	2 个 1 GB 以太网 NIC，建议 10 GB
ceph-radosgw-container	处理器	每个 radosgw-container 1 个 AMD64 或 Intel 64 CPU 内核
	RAM	每个守护进程 1 GB
	磁盘空间	每个守护进程 5 GB
	网络	1 个 1 GB 以太网 NIC
ceph-mds-container	处理器	每个 mds-container 1 个 AMD64 或 Intel 64 CPU 内核
	RAM	每个 **mds-container** 3 GB 这个数字高度依赖于可配置的 MDS 缓存大小。RAM 要求通常为 **mds_cache_memory_limit** 配置设置中设置的两倍。另请注意，这是守护进程的内存，而不是整体系统内存。
	磁盘空间	每个 **mds-container** 2 GB，并考虑可能调试日志所需的额外空间，20GB 是个不错的起点。
	网络	2 个 1 GB 以太网 NIC，建议 10 GB 请注意，这与 OSD 容器网络相同。如果您在 OSD 上有一个 10 GB 网络，则应在 MDS 上使用相同的网络，这样 MDS 在涉及延迟时不会产生判断。

2.9. 其它资源

- 如果要深入了解 Ceph 的各种内部组件，以及这些组件的相关策略，请参阅*Red Hat Ceph Storage 策略指南*以了解更多详细信息。

第 3 章 安装 RED HAT CEPH STORAGE 的要求

图 3.1. 先决条件工作流



在安装 Red Hat Ceph Storage 之前，请先查看以下要求，并相应地准备各个 monitor、OSD、元数据服务器和客户端节点。

注意

要了解 Red Hat Ceph Storage 版本和对应的 Red Hat Ceph Storage 软件包版本，请参阅红帽客户门户网站中的 Red Hat Ceph Storage 版本和对应的 Ceph 软件包版本 。

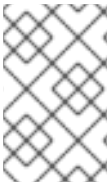
3.1. 先决条件

- 检查硬件是否满足 Red Hat Ceph Storage 4 的最低要求。

3.2. 安装 RED HAT CEPH STORAGE 的要求清单

Task	必填	节	建议
验证操作系统版本	是	第 3.3 节 “Red Hat Ceph Storage 的操作系统要求”	
注册 Ceph 节点	是	第 3.4 节 “将 Red Hat Ceph Storage 节点注册到 CDN 并附加订阅”	
启用 Ceph 软件存储库	是	第 3.5 节 “启用 Red Hat Ceph Storage 存储库”	
使用带有 OSD 节点的 RAID 控制器	否	第 2.6 节 “将 RAID 控制器用于 OSD 节点的注意事项”	在 RAID 控制器中启用回写缓存可能会导致 OSD 节点增加较小的 I/O 写入吞吐量。
配置网络	是	第 3.6 节 “验证 Red Hat Ceph Storage 的网络配置”	至少需要一个公共网络。但是，推荐使用一个专用的网络以用于集群通信。

Task	必填	节	建议
配置防火墙	否	第 3.7 节 “为 Red Hat Ceph Storage 配置防火墙”	防火墙可以提高网络的信任级别。
创建 Ansible 用户	是	第 3.8 节 “创建具有 **sudo** 访问权限的 Ansible 用户”	所有 Ceph 节点上都需要创建 Ansible 用户。
启用无密码 SSH	是	第 3.9 节 “为 Ansible 启用免密码 SSH”	Ansible 需要。



注意

默认情况下，**ceph-ansible** 会根据需要安装 NTP/chronyd。如果使用自定义 NTP/chronyd，请参阅手动安装 Red Hat Ceph Storage 中 的*为 Red Hat Ceph Storage 配置网络时间协议*部分，以了解必须如何配置 NTP/chronyd 以可以与 Ceph 正常运行。

3.3. RED HAT CEPH STORAGE 的操作系统要求

Red Hat Enterprise Linux 权利包括在 Red Hat Ceph Storage 订阅中。

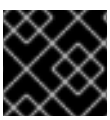
Red Hat Ceph Storage 4 的初始发行版本在 Red Hat Enterprise Linux 7.7 或 Red Hat Enterprise Linux 8.1 上被支持。Red Hat Ceph Storage 4.3 的当前版本在 Red Hat Enterprise Linux 7.9, 8.2 EUS, 8.4 EUS, 8.5, 8.6, 8.7, 8.8 上被支持。

基于 RPM 的部署或基于容器的部署支持 Red Hat Ceph Storage 4。

重要

在运行 Red Hat Enterprise Linux 7 的容器中部署 Red Hat Ceph Storage 4，部署运行在 Red Hat Enterprise Linux 8 容器镜像上的 Red Hat Ceph Storage 4。

在所有节点上使用相同的操作系统版本、架构和部署类型。例如，请勿将具有 AMD64 和 Intel 64 架构的节点混合使用，将节点与 Red Hat Enterprise Linux 7 和 Red Hat Enterprise Linux 8 操作系统混合，或使用基于 RPM 的部署和基于容器的部署混合节点。



重要

红帽不支持具有异构架构、操作系统版本或部署类型的集群。

SELinux

默认情况下，SELinux 设置为 **Enforcing** 模式，并且安装了 **ceph-selinux** 软件包。有关 SELinux 的更多信息，请参阅 *数据安全和强化指南*、*Red Hat Enterprise Linux 7 SELinux 用户和管理员指南*，以及使用 *SELinux 的* Red Hat Enterprise Linux 8。

其它资源

其它资源

- Red Hat Enterprise Linux 8 的文档包括在 https://access.redhat.com/documentation/zh-cn/red_hat_enterprise_linux/8/
- Red Hat Enterprise Linux 7 的文档包括在 https://access.redhat.com/documentation/zh-cn/red_hat_enterprise_linux/7/。

返回要求清单

3.4. 将 RED HAT CEPH STORAGE 节点注册到 CDN 并附加订阅

将每个 Red Hat Ceph Storage 节点注册到 Content Delivery Network (CDN)，再附加适当的订阅，以便节点可以访问软件存储库。每个 Red Hat Ceph Storage 节点都必须能够访问完整的 Red Hat Enterprise Linux 8 基本内容以及 extras 存储库内容。除非另有说明，否则在存储集群中的所有裸机和容器节点上执行以下步骤。

注意

对于在安装过程中无法访问互联网的裸机红帽 Ceph 存储节点，请使用 Red Hat Satellite 服务器提供软件内容。或者，挂载本地 Red Hat Enterprise Linux 8 服务器 ISO 镜像，并将 Red Hat Ceph Storage 节点指向 ISO 镜像。如需更多详细信息，请联系红帽支持。

有关将 Ceph 节点注册到 Red Hat Satellite 服务器的更多信息，请参阅 如何将 Ceph 注册到 Satellite 6，以及如何在红帽客户门户上将 Ceph 注册到 Satellite 5 文章。

先决条件

- 有效的红帽订阅。
- Red Hat Ceph Storage 节点必须能够连接互联网。
- 对 Red Hat Ceph Storage 节点的根级别访问权限.

流程

1. 仅用于**容器**部署，当 Red Hat Ceph Storage 节点在部署期间 **无法**访问互联网时。您必须首先在可访问互联网的节点中执行这些步骤：

   a. 启动本地容器 registry：

   Red Hat Enterprise Linux 7

   ```
   # docker run -d -p 5000:5000 --restart=always --name registry registry:2
   ```

   Red Hat Enterprise Linux 8

   ```
   # podman run -d -p 5000:5000 --restart=always --name registry registry:2
   ```

   b. 确定 **registry.redhat.io** 位于容器 registry 搜索路径中。
   打开以编辑 **/etc/containers/registries.conf** 文件：

   ```
   [registries.search]
   registries = [ 'registry.access.redhat.com', 'registry.fedoraproject.org',
   'registry.centos.org', 'docker.io']
   ```

如果文件中没有包括 **registry.redhat.io**，请添加它：

```
[registries.search]
registries = ['registry.redhat.io', 'registry.access.redhat.com', 'registry.fedoraproject.org',
'registry.centos.org', 'docker.io']
```

c. 从红帽客户门户网站拉取 Red Hat Ceph Storage 4 镜像、Prometheus 镜像和 Dashboard 镜像：

Red Hat Enterprise Linux 7

```
# docker pull registry.redhat.io/rhceph/rhceph-4-rhel8:latest
# docker pull registry.redhat.io/openshift4/ose-prometheus-node-exporter:v4.6
# docker pull registry.redhat.io/rhceph/rhceph-4-dashboard-rhel8:latest
# docker pull registry.redhat.io/openshift4/ose-prometheus:v4.6
# docker pull registry.redhat.io/openshift4/ose-prometheus-alertmanager:v4.6
```

Red Hat Enterprise Linux 8

```
# podman pull registry.redhat.io/rhceph/rhceph-4-rhel8:latest
# podman pull registry.redhat.io/openshift4/ose-prometheus-node-exporter:v4.6
# podman pull registry.redhat.io/rhceph/rhceph-4-dashboard-rhel8:latest
# podman pull registry.redhat.io/openshift4/ose-prometheus:v4.6
# podman pull registry.redhat.io/openshift4/ose-prometheus-alertmanager:v4.6
```

注意

Red Hat Enterprise Linux 7 和 8 都使用相同的容器镜像，它们基于 Red Hat Enterprise Linux 8。

d. 标记镜像：
Prometheus 镜像标签版本是 Red Hat Ceph Storage 4.2 的 v4.6。

Red Hat Enterprise Linux 7

```
 # docker tag registry.redhat.io/rhceph/rhceph-4-rhel8:latest
LOCAL_NODE_FQDN:5000/rhceph/rhceph-4-rhel8:latest
 # docker tag registry.redhat.io/openshift4/ose-prometheus-node-exporter:v4.6
LOCAL_NODE_FQDN:5000/openshift4/ose-prometheus-node-exporter:v4.6
 # docker tag registry.redhat.io/rhceph/rhceph-4-dashboard-rhel8:latest
LOCAL_NODE_FQDN:5000/rhceph/rhceph-4-dashboard-rhel8:latest
 # docker tag registry.redhat.io/openshift4/ose-prometheus-alertmanager:v4.6
LOCAL_NODE_FQDN:5000/openshift4/ose-prometheus-alertmanager:v4.6
 # docker tag registry.redhat.io/openshift4/ose-prometheus:v4.6
LOCAL_NODE_FQDN:5000/openshift4/ose-prometheus:v4.6
```

替换

- *LOCAL_NODE_FQDN*，使用您的本地主机 FQDN。

Red Hat Enterprise Linux 8

```
 # podman tag registry.redhat.io/rhceph/rhceph-4-rhel8:latest
LOCAL_NODE_FQDN:5000/rhceph/rhceph-4-rhel8:latest
 # podman tag registry.redhat.io/openshift4/ose-prometheus-node-exporter:v4.6
LOCAL_NODE_FQDN:5000/openshift4/ose-prometheus-node-exporter:v4.6
 # podman tag registry.redhat.io/rhceph/rhceph-4-dashboard-rhel8:latest
LOCAL_NODE_FQDN:5000/rhceph/rhceph-4-dashboard-rhel8:latest
 # podman tag registry.redhat.io/openshift4/ose-prometheus-alertmanager:v4.6
LOCAL_NODE_FQDN:5000/openshift4/ose-prometheus-alertmanager:v4.6
 # podman tag registry.redhat.io/openshift4/ose-prometheus:v4.6
LOCAL_NODE_FQDN:5000/openshift4/ose-prometheus:v4.6
```

替换

- *LOCAL_NODE_FQDN*，使用您的本地主机 FQDN。

e. 编辑 **/etc/containers/registries.conf** 文件，在文件中添加带有端口信息的节点 FQDN，并保存：

```
[registries.insecure]
registries = ['LOCAL_NODE_FQDN:5000']
```

注意

必须在访问本地 Docker registry 的所有存储集群节点上执行此步骤。

f. 将镜像推送到您启动的本地 Docker registry：

Red Hat Enterprise Linux 7

```
 # docker push --remove-signatures LOCAL_NODE_FQDN:5000/rhceph/rhceph-4-rhel8
 # docker push --remove-signatures LOCAL_NODE_FQDN:5000/openshift4/ose-
prometheus-node-exporter:v4.6
 # docker push --remove-signatures LOCAL_NODE_FQDN:5000/rhceph/rhceph-4-
dashboard-rhel8
 # docker push --remove-signatures LOCAL_NODE_FQDN:5000/openshift4/ose-
prometheus-alertmanager:v4.6
 # docker push --remove-signatures LOCAL_NODE_FQDN:5000/openshift4/ose-
prometheus:v4.6
```

替换

- *LOCAL_NODE_FQDN*，使用您的本地主机 FQDN。

Red Hat Enterprise Linux 8

```
 # podman push --remove-signatures LOCAL_NODE_FQDN:5000/rhceph/rhceph-4-rhel8
 # podman push --remove-signatures LOCAL_NODE_FQDN:5000/openshift4/ose-
prometheus-node-exporter:v4.6
 # podman push --remove-signatures LOCAL_NODE_FQDN:5000/rhceph/rhceph-4-
dashboard-rhel8
 # podman push --remove-signatures LOCAL_NODE_FQDN:5000/openshift4/ose-
```

```
prometheus-alertmanager:v4.6
 # podman push --remove-signatures LOCAL_NODE_FQDN:5000/openshift4/ose-
prometheus:v4.6
```

替换

- *LOCAL_NODE_FQDN*，使用您的本地主机 FQDN。

g. 对于 Red Hat Enterprise Linux 7，重启 **docker** 服务：

```
# systemctl restart docker
```

注意

当 Red Hat Ceph Storage 节点在部署过程中无法访问互联网时，请参阅 *安装 Red Hat Ceph Storage 集群*来了解 **all.yml** 文件的示例。

2. 对于所有部署，**裸机或容器中** ：

a. 注册该节点，并在提示时输入适当的红帽客户门户网站凭证：

```
# subscription-manager register
```

b. 从 CDN 拉取最新的订阅数据：

```
# subscription-manager refresh
```

c. 列出 Red Hat Ceph Storage 的所有可用订阅：

```
# subscription-manager list --available --all --matches="*Ceph*"
```

从 Red Hat Ceph Storage 可用订阅列表中复制池 ID。

d. 附加订阅：

```
# subscription-manager attach --pool=POOL_ID
```

替换

- 带有上一步中标识的池 ID 的 *POOL_ID*。

e. 禁用默认软件存储库，并在相应版本的 Red Hat Enterprise Linux 中启用服务器和附加软件仓库：

Red Hat Enterprise Linux 7

```
# subscription-manager repos --disable=*
# subscription-manager repos --enable=rhel-7-server-rpms
# subscription-manager repos --enable=rhel-7-server-extras-rpms
```

Red Hat Enterprise Linux 8

```
# subscription-manager repos --disable=*
# subscription-manager repos --enable=rhel-8-for-x86_64-baseos-rpms
# subscription-manager repos --enable=rhel-8-for-x86_64-appstream-rpms
```

3. 更新系统，以接收最新的软件包。

   a. Red Hat Enterprise Linux 7：

   ```
   # yum update
   ```

   b. Red Hat Enterprise Linux 8：

   ```
   # dnf update
   ```

其它资源

- 有关红帽订阅管理，请参阅 *使用和配置红帽订阅管理器指南*。
- 请参阅 *启用 Red Hat Ceph Storage 存储库*。

返回要求清单

3.5. 启用 RED HAT CEPH STORAGE 存储库

您必须先选择安装方法，然后才能安装 Red Hat Ceph Storage。Red Hat Ceph Storage 支持两种安装方法：

- 内容交付网络 (CDN)
  对于带有可以直接连接到互联网的 Ceph 节点的 Ceph 存储集群，请使用红帽订阅管理器来启用所需的 Ceph 存储库。
- 本地存储库
  对于安全措施使节点无法访问互联网的 Ceph 存储集群，请从作为 ISO 镜像提供的单个软件构建中安装 Red Hat Ceph Storage 4，这将允许您安装本地存储库。

先决条件

- 有效的客户订阅。
- 对于 CDN 安装：
  - Red Hat Ceph Storage 节点必须能够连接互联网。
  - 使用 CDN 注册集群节点。
- 如果启用，请禁用 Extra Packages for Enterprise Linux (EPEL) 软件存储库：

```
[root@monitor ~]# yum install yum-utils vim -y
[root@monitor ~]# yum-config-manager --disable epel
```

流程

- 对于 CDN 安装：
  在 **Ansible 管理节点上**，启用 Red Hat Ceph Storage 4 Tools 存储库和 Ansible 存储库：

Red Hat Enterprise Linux 7

```
[root@admin ~]# subscription-manager repos --enable=rhel-7-server-rhceph-4-tools-rpms --enable=rhel-7-server-ansible-2.9-rpms
```

Red Hat Enterprise Linux 8

```
[root@admin ~]# subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms --enable=ansible-2.9-for-rhel-8-x86_64-rpms
```

- 默认情况下，**ceph-ansible** 在对应的节点上启用红帽 Ceph 存储存储库。手动启用软件仓库：

注意

不要在容器化部署中启用这些存储库，因为不需要它们。

在 **Ceph 监控节点上**，启用 Red Hat Ceph Storage 4 monitor 存储库：

Red Hat Enterprise Linux 7

```
[root@monitor ~]# subscription-manager repos --enable=rhel-7-server-rhceph-4-mon-rpms
```

Red Hat Enterprise Linux 8

```
[root@monitor ~]# subscription-manager repos  --enable=rhceph-4-mon-for-rhel-8-x86_64-rpms
```

在 **Ceph OSD 节点上**，启用 Red Hat Ceph Storage 4 OSD 存储库：

Red Hat Enterprise Linux 7

```
[root@osd ~]# subscription-manager repos --enable=rhel-7-server-rhceph-4-osd-rpms
```

Red Hat Enterprise Linux 8

```
[root@osd ~]# subscription-manager repos --enable=rhceph-4-osd-for-rhel-8-x86_64-rpms
```

在以下节点类型上启用 Red Hat Ceph Storage 4 工具存储库：**RBD mirroring**、**Ceph clients**、**Ceph Object Gateways**、**Metadata Servers**、**NFS**、**iSCSI gateways** 和 **Dashboard servers**

Red Hat Enterprise Linux 7

```
[root@client ~]# subscription-manager repos --enable=rhel-7-server-rhceph-4-tools-rpms
```

Red Hat Enterprise Linux 8

```
[root@client ~]# subscription-manager repos  --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
```

- 对于 ISO 安装：

  1. 登录红帽客户门户。
  2. 点 **Downloads** 访问 **Software & Download**Center。
  3. 在 Red Hat Ceph Storage 区域中，单击 **Download Software** 以下载最新版本的软件。

其它资源

- Red Hat Subscription Management 1 使用和配置 Red Hat Subscription Manager 指南

返回要求清单

3.6. 验证 RED HAT CEPH STORAGE 的网络配置

所有 Red Hat Ceph Storage 节点都需要一个公共网络。您必须具有一个网络接口卡，配置为一个公共网络，Ceph 客户端可以访问 Ceph 监视器和 Ceph OSD 节点。

您可能有一个用于集群网络的网络接口卡，以便 Ceph 可以在独立于公共网络的网络上执行心跳、对等、复制和恢复。

配置网络接口设置，并确保这些更改永久保留。

重要

红帽不推荐为公共和专用网络使用单个网络接口卡。

先决条件

- 连接到网络的网络接口卡。

流程

以 **root** 用户身份，在存储集群中的所有 Red Hat Ceph Storage 节点上执行以下步骤。

1. 验证对应于面向公共的网络接口卡的 **/etc/sysconfig/network-scripts/ifcfg-*** 文件中是否有以下设置：

   a. 对于静态 IP 地址，把 **BOOTPROTO** 参数设置为 **none**。

   b. **ONBOOT** 参数必须设置为 **yes**。
   如果设为 **no**，Ceph 存储集群在重启后可能无法成为 peer。

   c. 如果要使用 IPv6，您必须将 IPv6 参数（如 **IPV6INIT** ）设置为 **yes**，**IPV6_FAILURE_FATAL** 参数除外。
   此外，编辑 Ceph 配置文件 **/etc/ceph/ceph.conf**，以指示 Ceph 使用 IPv6，否则 Ceph 会使用 IPv4。

其它资源

- 有关为 Red Hat Enterprise Linux 8 配置网络接口脚本的详情，请参考为 Red Hat Enterprise Linux 8 *配置和管理网络*指南中的*使用 ifcfg 文件配置 ip 网络*一章。
- 如需有关网络配置的更多信息，请参阅《红帽 *Ceph 存储 4 配置指南》 中的 Ceph 网络配置* 章节。

返回要求清单

3.7. 为 RED HAT CEPH STORAGE 配置防火墙

Red Hat Ceph Storage 使用 **firewalld** 服务。**firewalld** 服务包含每个守护进程的端口列表。

Ceph Monitor 守护进程使用端口 **3300** 和 **6789** 作为 Ceph 存储集群中的通信。

在每个 Ceph OSD 节点上，OSD 守护进程使用 **6800-7300** 范围内的多个端口：

- 一个用于通过公共网络与客户端通信和监控器
- 一个用于通过集群网络发送数据到其他 OSD（如果可用）；否则，通过公共网络发送数据
- 一个用于通过集群网络（如果有）交换心跳数据包；否则，通过公共网络交换。

Ceph 管理器 (**ceph-mgr**) 守护进程使用范围为 **6800-7300** 的端口。考虑将 **ceph-mgr** 守护进程与 Ceph monitor 在同一节点上并置。

Ceph 元数据服务器节点 (**ceph-mds**) 使用端口范围 **6800-7300**。

Ceph 对象网关节点由 Ansible 配置为使用默认端口 **8080**。但是，您可以更改默认端口，例如端口 **80**。

要使用 SSL/TLS 服务，请打开端口 **443**。

如果启用 **firewalld**，以下步骤是可选的。默认情况下，**ceph-ansible** 在 **group_vars/all.yml** 中包括以下设置，它会自动打开适当的端口：

```
configure_firewall: True
```

前提条件

- 网络硬件已连接。
- 对存储集群中的所有节点具有 **root** 或 **sudo** 访问权限。

流程

1. 在存储集群的所有节点上，启动 **firewalld** 服务。启用它在引导时运行，并确保它正在运行：

```
# systemctl enable firewalld
# systemctl start firewalld
# systemctl status firewalld
```

2. 在所有监控节点上，在公共网络上打开端口 **3300** 和 **6789**：

```
[root@monitor ~]# firewall-cmd --zone=public --add-port=3300/tcp
[root@monitor ~]# firewall-cmd --zone=public --add-port=3300/tcp --permanent
[root@monitor ~]# firewall-cmd --zone=public --add-port=6789/tcp
[root@monitor ~]# firewall-cmd --zone=public --add-port=6789/tcp --permanent
[root@monitor ~]# firewall-cmd --permanent --add-service=ceph-mon
[root@monitor ~]# firewall-cmd --add-service=ceph-mon
```

根据源地址限制访问：

-

```
firewall-cmd --zone=public --add-rich-rule='rule family=ipv4 \
source address=IP_ADDRESS/NETMASK_PREFIX port protocol=tcp \
port=6789 accept' --permanent
```

替换

- *IP_ADDRESS*，使用 Monitor 节点的网络地址。
- *NETMASK_PREFIX*，使用 CIDR 表示法的子网掩码 。

示例

```
[root@monitor ~]# firewall-cmd --zone=public --add-rich-rule='rule family=ipv4 \
source address=192.168.0.11/24 port protocol=tcp \
port=6789 accept' --permanent
```

3. 在所有 OSD 节点上，打开公共网络上的端口 **6800-7300** ：

```
[root@osd ~]# firewall-cmd --zone=public --add-port=6800-7300/tcp
[root@osd ~]# firewall-cmd --zone=public --add-port=6800-7300/tcp --permanent
[root@osd ~]# firewall-cmd --permanent --add-service=ceph
[root@osd ~]# firewall-cmd --add-service=ceph
```

如果您有单独的集群网络，请对适当的区重复这些命令。

4. 在所有 Ceph Manager (**ceph-mgr**) 节点上，打开公共网络上的端口 **6800-7300** ：

```
[root@monitor ~]# firewall-cmd --zone=public --add-port=6800-7300/tcp
[root@monitor ~]# firewall-cmd --zone=public --add-port=6800-7300/tcp --permanent
```

如果您有单独的集群网络，请对适当的区重复这些命令。

5. 在所有 Ceph 元数据服务器 (**ceph-mds**) 节点上，打开公共网络上的端口 **6800-7300** ：

```
[root@monitor ~]# firewall-cmd --zone=public --add-port=6800-7300/tcp
[root@monitor ~]# firewall-cmd --zone=public --add-port=6800-7300/tcp --permanent
```

如果您有单独的集群网络，请对适当的区重复这些命令。

6. 在所有 Ceph 对象网关节点上，打开公共网络上的相关端口或端口。

   a. 打开默认 Ansible 配置的端口 **8080**:

```
[root@gateway ~]# firewall-cmd --zone=public --add-port=8080/tcp
[root@gateway ~]# firewall-cmd --zone=public --add-port=8080/tcp --permanent
```

   根据源地址限制访问：

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_ADDRESS/NETMASK_PREFIX" port protocol="tcp" \
port="8080" accept"
```

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_ADDRESS/NETMASK_PREFIX" port protocol="tcp" \
port="8080" accept" --permanent
```

替换

- *IP_ADDRESS*，使用 Monitor 节点的网络地址。
- *NETMASK_PREFIX*，使用 CIDR 表示法的子网掩码 。

示例

```
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4"
\
source address="192.168.0.31/24" port protocol="tcp" \
port="8080" accept"
```

```
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4"
\
source address="192.168.0.31/24" port protocol="tcp" \
port="8080" accept" --permanent
```

b. 另外，如果您使用 Ansible 安装 Ceph 对象网关并更改了 Ansible 配置 Ceph 对象网关的默认端口以便从 **8080** 使用，例如端口 **80**，打开此端口：

```
[root@gateway ~]# firewall-cmd --zone=public --add-port=80/tcp
[root@gateway ~]# firewall-cmd --zone=public --add-port=80/tcp --permanent
```

要根据源地址限制访问，请运行以下命令：

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_ADDRESS/NETMASK_PREFIX" port protocol="tcp" \
port="80" accept"
```

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_ADDRESS/NETMASK_PREFIX" port protocol="tcp" \
port="80" accept" --permanent
```

替换

- *IP_ADDRESS*，使用 Monitor 节点的网络地址。
- *NETMASK_PREFIX*，使用 CIDR 表示法的子网掩码 。

示例

```
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.31/24" port protocol="tcp" \
port="80" accept"
```

```
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.31/24" port protocol="tcp" \
port="80" accept" --permanent
```

c. 可选。要使用 SSL/TLS，请打开端口 **443** ：

```
[root@gateway ~]# firewall-cmd --zone=public --add-port=443/tcp
[root@gateway ~]# firewall-cmd --zone=public --add-port=443/tcp --permanent
```

要根据源地址限制访问，请运行以下命令：

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_ADDRESS/NETMASK_PREFIX" port protocol="tcp" \
port="443" accept"
```

```
firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="IP_ADDRESS/NETMASK_PREFIX" port protocol="tcp" \
port="443" accept" --permanent
```

替换

- *IP_ADDRESS*，使用 Monitor 节点的网络地址。
- *NETMASK_PREFIX*，使用 CIDR 表示法的子网掩码 。

示例

```
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.31/24" port protocol="tcp" \
port="443" accept"
[root@gateway ~]# firewall-cmd --zone=public --add-rich-rule="rule family="ipv4" \
source address="192.168.0.31/24" port protocol="tcp" \
port="443" accept" --permanent
```

其它资源

- 如需有关公共和集群网络的更多信息，请参阅验证 Red Hat Ceph Storage 的网络配置 。
- 有关 **firewalld** 的详情，请参阅 Red Hat Enterprise Linux 8 *网络安全*指南中的使用和配置防火墙章节。

返回要求清单

3.8. 创建具有 SUDO 访问权限的 ANSIBLE 用户

Ansible 必须能够以具有 **root** 权限的用户身份登录所有 Red Hat Ceph Storage 节点，以便在不提示输入密码的情况下安装软件和创建配置文件。在使用 Ansible 部署和配置 Red Hat Ceph Storage 集群时，您必须在存储集群的所有节点上创建一个没有密码的 **root** 用户。

前提条件

- 对存储集群中的所有节点具有 **root** 或 **sudo** 访问权限。

流程

1. 以 **root** 用户身份登录节点：

```
ssh root@HOST_NAME
```

替换

- *HOST_NAME*，使用 Ceph 节点的主机名。

示例

```
# ssh root@mon01
```

出现提示时，输入 **root** 密码。

2. 创建一个新的 Ansible 用户：

```
adduser USER_NAME
```

替换

- *USER_NAME*，使用具有 Ansible 用户的新用户名。

示例

```
# adduser admin
```

重要

不要使用 **ceph** 作为用户名。**ceph** 用户名保留用于 Ceph 守护进程。整个集群中的统一用户名可以提高易用性，但避免使用明显的用户名，因为入侵者通常使用它们进行暴力攻击。

3. 为这个用户设置一个新密码：

```
# passwd USER_NAME
```

替换

- *USER_NAME*，使用具有 Ansible 用户的新用户名。

示例

```
# passwd admin
```

出现提示时，输入新密码两次。

4. 为新创建的用户配置 **sudo** 访问权限：

```
cat << EOF >/etc/sudoers.d/USER_NAME
$USER_NAME ALL = (root) NOPASSWD:ALL
EOF
```

替换

- *USER_NAME*，使用具有 Ansible 用户的新用户名。

  示例

  ```
  # cat << EOF >/etc/sudoers.d/admin
  admin ALL = (root) NOPASSWD:ALL
  EOF
  ```

5. 为新文件分配正确的文件权限：

```
chmod 0440 /etc/sudoers.d/USER_NAME
```

替换

- *USER_NAME*，使用具有 Ansible 用户的新用户名。

  示例

  ```
  # chmod 0440 /etc/sudoers.d/admin
  ```

其它资源

- Red Hat Enterprise Linux 8 的*配置基本系统设置* 指南中的管理用户帐户部分

返回要求清单

3.9. 为 ANSIBLE 启用免密码 SSH

在 Ansible 管理节点上生成 SSH 密钥对，并将公钥分发到存储集群中的每个节点，以便 Ansible 可以在不提示输入密码的情况下访问节点。

注意

如果使用基于 Cockpit Web 的界面安装 Red Hat Ceph Storage，则不需要这个过程。这是因为 Cockpit Ceph 安装程序会生成自己的 SSH 密钥。有关将 Cockpit SSH 密钥复制到集群中的所有节点的说明，请参见使用 Cockpit Web 界面安装 Red Hat Ceph Storage 一章。

先决条件

- 访问 Ansible 管理节点.
- *创建具有 **sudo** 访问权限的 Ansible 用户*.

流程

1. 生成 SSH 密钥对，接受默认文件名并将密语留空：

```
[ansible@admin ~]$ ssh-keygen
```

2. 将公钥复制到存储集群中的所有节点：

```
ssh-copy-id USER_NAME@HOST_NAME
```

替换

- *USER_NAME*，使用具有 Ansible 用户的新用户名。
- *HOST_NAME*，使用 Ceph 节点的主机名。

示例

```
[ansible@admin ~]$ ssh-copy-id ceph-admin@ceph-mon01
```

3. 创建用户的 SSH **config** 文件 ：

```
[ansible@admin ~]$ touch ~/.ssh/config
```

4. 打开并编辑 **config** 文件。为存储集群中每个节点的 **Hostname** 和 **User** 选项设置值：

```
Host node1
  Hostname HOST_NAME
  User USER_NAME
Host node2
  Hostname HOST_NAME
  User USER_NAME
...
```

替换

- *HOST_NAME*，使用 Ceph 节点的主机名。
- *USER_NAME*，使用具有 Ansible 用户的新用户名。

示例

```
Host node1
  Hostname monitor
  User admin
Host node2
  Hostname osd
  User admin
Host node3
  Hostname gateway
  User admin
```

重要

通过配置 **~/.ssh/config** 文件，您不必在每次执行 **ansible-playbook** 命令时指定 **-u *USER_NAME*** 选项。

5. 为 **~/.ssh/config** 文件设置正确的文件权限：

```
[admin@admin ~]$ chmod 600 ~/.ssh/config
```

其它资源

- **ssh_config(5)** 手册页面。
- 请参阅 Red Hat Enterprise Linux 8 *安全网络*中的使用 OpenSSH 在两个系统间使用安全通信一章。

返回要求清单

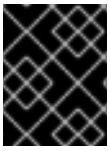
第 4 章 使用 COCKPIT WEB 界面安装 RED HAT CEPH STORAGE

本章论述了如何使用 Cockpit Web 界面安装 Red Hat Ceph Storage 集群和其他组件，如元数据服务器、Ceph 客户端或 Ceph 对象网关。

此过程包括安装 Cockpit Ceph Installer、登录 Cockpit，以及使用安装程序内的不同页面配置和启动集群安装。

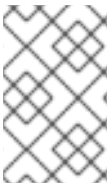
注意

Cockpit Ceph 安装程序使用 Ansible 和 **ceph-ansible** RPM 提供的 Ansible playbook 来执行实际安装。仍可以使用这些 playbook 在不使用 Cockpit 的情况下安装 Ceph。这个过程与本章相关，称为 *Ansible 直接安装*，或者 *直接使用 Ansible playbook*。



重要

Cockpit Ceph 安装程序目前不支持 IPv6 网络。如果需要 IPv6 网络，请直接使用 Ansible playbook 安装 Ceph。



注意

默认情况下，用于管理和监控 Ceph 的控制面板 Web 界面由 **ceph-ansible** RPM 中的 Ansible playbook 安装，Cockpit 在后端上使用它。因此，无论您直接使用 Ansible playbook，还是使用 Cockpit 安装 Ceph，也将安装控制面板 Web 界面。

4.1. 先决条件

- 完成直接安装 Ansible Red Hat Ceph Storage 所需的 一般先决条件。
- Firefox 或 Chrome 的最新版本。
- 如果使用多个网络来分段集群内流量、客户端至集群流量、RADOS 网关流量或 iSCSI 流量，请确保主机上已配置了相关的网络。如需更多信息，请参阅硬件指南中的网络注意事项，以及本章中有关完成 Cockpit Ceph 安装程序的网络页面 的章节。
- 确保 Cockpit Web 界面的默认端口 **9090** 可以访问。

4.2. 安装要求

- 一个节点充当 Ansible 管理节点。
- 一个提供性能指标和警报平台的节点。这可以与 Ansible 管理节点在一起。
- 组成 Ceph 集群的一个或多个节点。安装程序支持名为 *Development/POC* 的一体化安装。在这种模式中，所有 Ceph 服务可以从同一节点运行，数据复制默认为磁盘而不是主机级别保护。

4.3. 安装和配置 COCKPIT CEPH 安装程序

在使用 Cockpit Ceph 安装程序安装 Red Hat Ceph Storage 集群之前，您必须在 Ansible 管理节点上安装 Cockpit Ceph 安装程序。

先决条件

- 对 Ansible 管理节点的根级别访问权限.
- 用于 Ansible 应用的 **ansible** 用户帐户。

流程

1. 验证已安装了 Cockpit。

```
$ rpm -q cockpit
```

例如：

```
[admin@jb-ceph4-admin ~]$ rpm -q cockpit
cockpit-196.3-1.el8.x86_64
```

如果您看到与上例类似的输出，请跳到*验证 Cockpit 正在运行*的步骤。如果输出是 **package cockpit is not installed**，继续*安装 Cockpit 步骤*。

2. 可选：安装 Cockpit。

   a. Red Hat Enterprise Linux 8：

   ```
   # dnf install cockpit
   ```

   b. Red Hat Enterprise Linux 7：

   ```
   # yum install cockpit
   ```

3. 验证 Cockpit 正在运行。

```
# systemctl status cockpit.socket
```

如果您在输出中看到 **Active: active (listening)**，请跳到*为 Red Hat Ceph Storage 安装 Cockpit 插件*的步骤。如果您看到 **Active: inactive(dead)**，请继续执行*启用 Cockpit 步骤*。

4. 可选：启用 Cockpit。

   a. 使用 **systemctl** 命令启用 Cockpit：

   ```
   # systemctl enable --now cockpit.socket
   ```

   您将看到类似如下的行：

   ```
   Created symlink /etc/systemd/system/sockets.target.wants/cockpit.socket →
   /usr/lib/systemd/system/cockpit.socket.
   ```

   b. 验证 Cockpit 是否正在运行：

   ```
   # systemctl status cockpit.socket
   ```

   您将看到类似如下的行：

   ```
   Active: active (listening) since Tue 2020-01-07 18:49:07 EST; 7min ago
   ```

5. 安装 Red Hat Ceph Storage 的 Cockpit Ceph 安装程序。

   a. Red Hat Enterprise Linux 8：

   ```
   # dnf install cockpit-ceph-installer
   ```

   b. Red Hat Enterprise Linux 7：

   ```
   # yum install cockpit-ceph-installer
   ```

6. 以 Ansible 用户身份，使用 sudo 登录容器目录：

   注意

   默认情况下，Cockpit Ceph 安装程序使用 **root** 用户安装 Ceph。若要使用作为安装 Ceph 的先决条件一部分创建的 Ansible 用户，请以 Ansible 用户身份通过 **sudo** 运行此流程中的其余命令。

   Red Hat Enterprise Linux 7

   ```
   $ sudo docker login -u CUSTOMER_PORTAL_USERNAME https://registry.redhat.io
   ```

   示例

   ```
   [admin@jb-ceph4-admin ~]$ sudo docker login -u myusername https://registry.redhat.io
   Password:
   Login Succeeded!
   ```

   Red Hat Enterprise Linux 8

   ```
   $ sudo podman login -u CUSTOMER_PORTAL_USERNAME https://registry.redhat.io
   ```

   示例

   ```
   [admin@jb-ceph4-admin ~]$ sudo podman login -u myusername https://registry.redhat.io
   Password:
   Login Succeeded!
   ```

7. 确定 **registry.redhat.io** 位于容器 registry 搜索路径中。

   a. 打开以编辑 **/etc/containers/registries.conf** 文件：

   ```
   [registries.search]
   registries = [ 'registry.access.redhat.com', 'registry.fedoraproject.org',
   'registry.centos.org', 'docker.io']
   ```

   如果文件中没有包括 **registry.redhat.io**，请添加它：

   ```
   [registries.search]
   registries = ['registry.redhat.io', 'registry.access.redhat.com', 'registry.fedoraproject.org',
   'registry.centos.org', 'docker.io']
   ```

8. 以 Ansible 用户身份，使用 sudo 启动 **ansible-runner-service**。

```
$ sudo ansible-runner-service.sh -s
```

示例

```
[admin@jb-ceph4-admin ~]$ sudo ansible-runner-service.sh -s
Checking environment is ready
Checking/creating directories
Checking SSL certificate configuration
Generating RSA private key, 4096 bit long modulus (2 primes)
.....
.....+++
.....+++
e is 65537 (0x010001)
Generating RSA private key, 4096 bit long modulus (2 primes)
.....+++
.....
.....+++
e is 65537 (0x010001)
writing RSA key
Signature ok
subject=C = US, ST = North Carolina, L = Raleigh, O = Red Hat, OU = RunnerServer, CN =
jb-ceph4-admin
Getting CA Private Key
Generating RSA private key, 4096 bit long modulus (2 primes)
.....+++
..+++
e is 65537 (0x010001)
writing RSA key
Signature ok
subject=C = US, ST = North Carolina, L = Raleigh, O = Red Hat, OU = RunnerClient, CN = jb-
ceph4-admin
Getting CA Private Key
Setting ownership of the certs to your user account(admin)
Setting target user for ansible connections to admin
Applying SELINUX container_file_t context to '/etc/ansible-runner-service'
Applying SELINUX container_file_t context to '/usr/share/ceph-ansible'
Ansible API (runner-service) container set to rhceph/ansible-runner-rhel8:latest
Fetching Ansible API container (runner-service). Please wait...
Trying to pull registry.redhat.io/rhceph/ansible-runner-rhel8:latest...Getting image source
signatures
Copying blob c585fd5093c6 done
Copying blob 217d30c36265 done
Copying blob e61d8721e62e done
Copying config b96067ea93 done
Writing manifest to image destination
Storing signatures
b96067ea93c8d6769eaea86854617c63c61ea10c4ff01ecf71d488d5727cb577
Starting Ansible API container (runner-service)
Started runner-service container
Waiting for Ansible API container (runner-service) to respond
The Ansible API container (runner-service) is available and responding to requests

Login to the cockpit UI at https://jb-ceph4-admin:9090/cockpit-ceph-installer to start the install
```

输出的最后一行包含 Cockpit Ceph 安装程序的 URL。在上例中，URL 是 **https://jb-ceph4-admin:9090/cockpit-ceph-installer**。记录下在环境中输出的 URL。

4.4. 将 COCKPIT CEPH INSTALLER SSH 密钥复制到集群中的所有节点

Cockpit Ceph 安装程序使用 SSH 连接并配置集群中的节点。为了让安装程序自动执行此操作，安装程序将生成 SSH 密钥对，使其可以在不提示输入密码的情况下访问节点。SSH 公钥必须传输到集群中的所有节点。

先决条件

- 已创建具有 sudo 访问权限的 Ansible 用户 。
- 已安装并配置了 Cockpit Ceph 安装程序。

流程

1. 以 Ansible 用户身份登录 Ansible 管理节点。

```
ssh ANSIBLE_USER@HOST_NAME
```

例如：

```
$ ssh admin@jb-ceph4-admin
```

2. 将 SSH 公钥复制到第一个节点：

```
sudo ssh-copy-id -f -i /usr/share/ansible-runner-service/env/ssh_key.pub
_ANSIBLE_USER_@_HOST_NAME_
```

例如：

```
$ sudo ssh-copy-id -f -i /usr/share/ansible-runner-service/env/ssh_key.pub admin@jb-ceph4-mon
/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/usr/share/ansible-runner-service/env/ssh_key.pub"
admin@192.168.122.182's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'admin@jb-ceph4-mon'"
and check to make sure that only the key(s) you wanted were added.
```

对集群中的所有节点重复此步骤

4.5. 登录 COCKPIT

您可以通过登录 Cockpit 来查看 Cockpit。

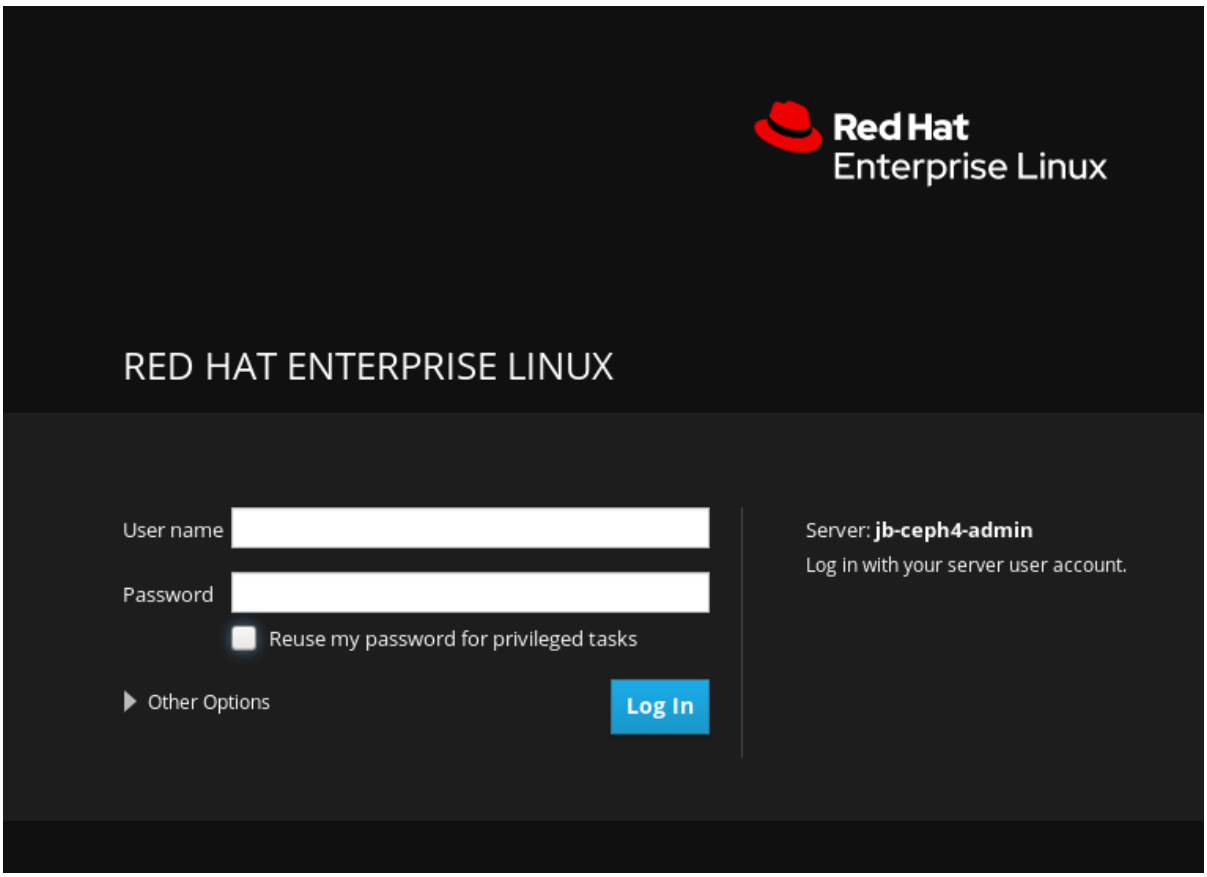
先决条件

- 已安装并配置了 Cockpit Ceph 安装程序。

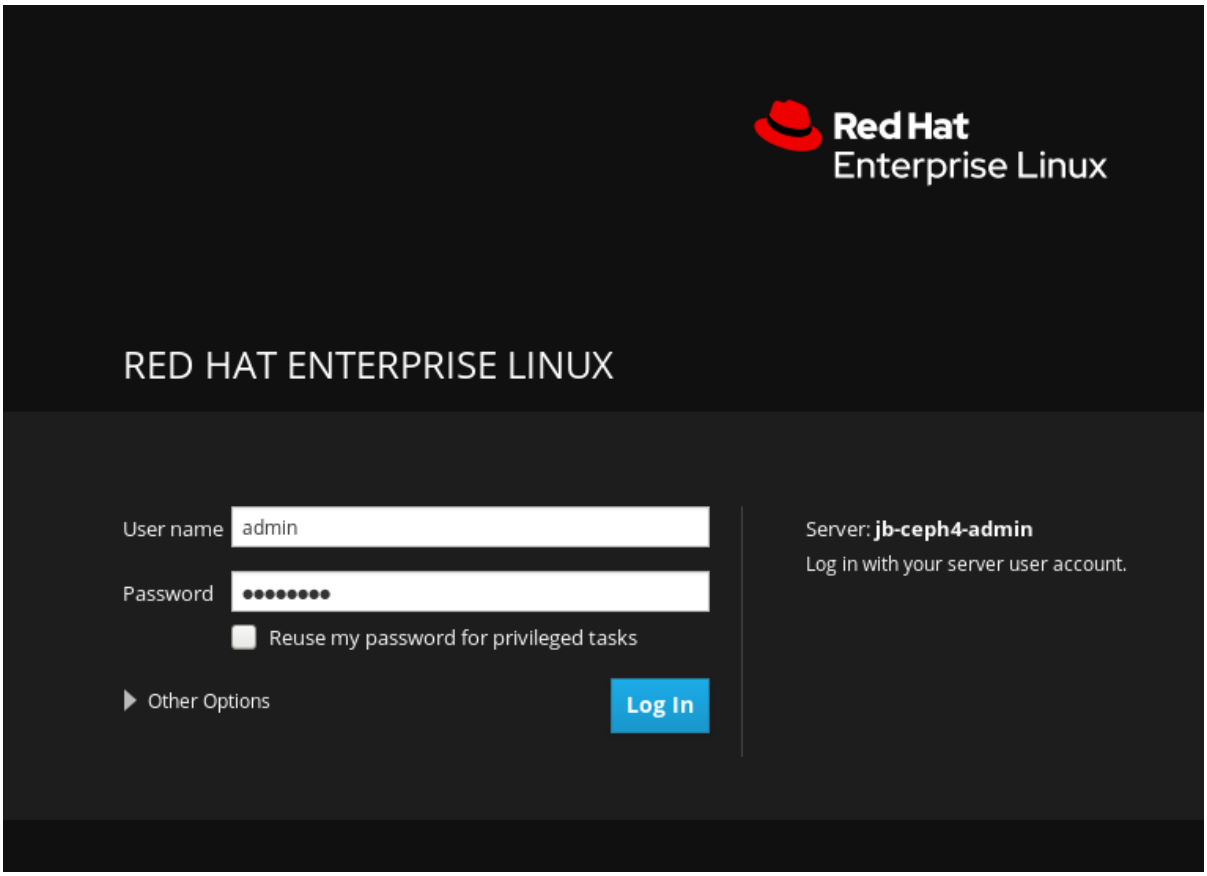
- 您有作为配置 Cockpit Ceph Installer 的一部分打印的 URL

流程

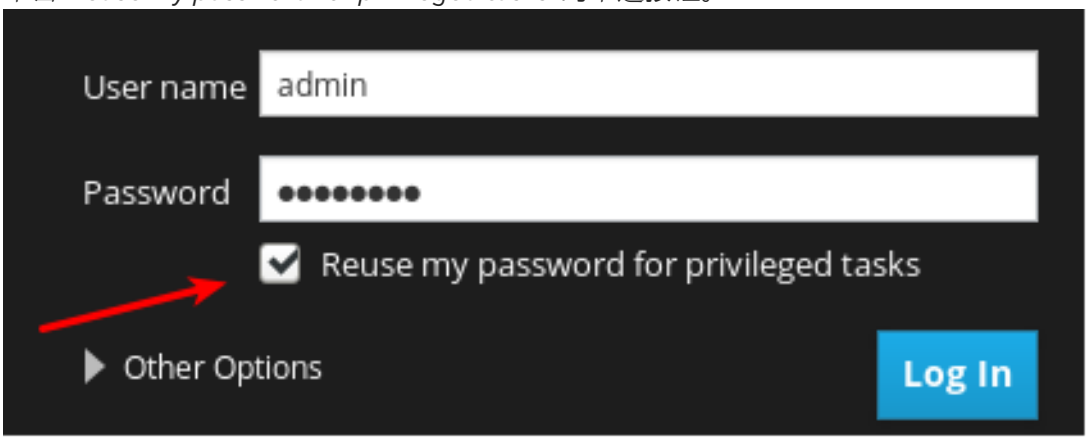
1. 在 Web 浏览器中打开 URL。



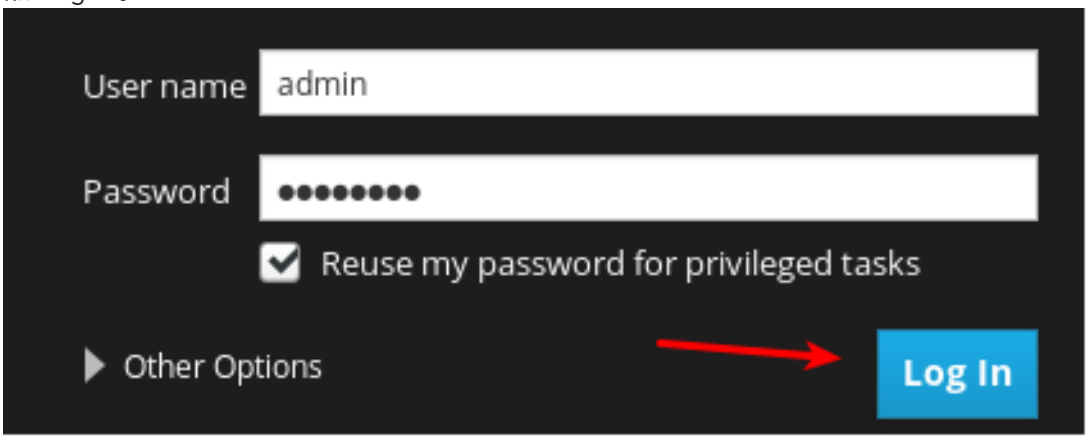
2. 输入 Ansible 用户名及其密码。



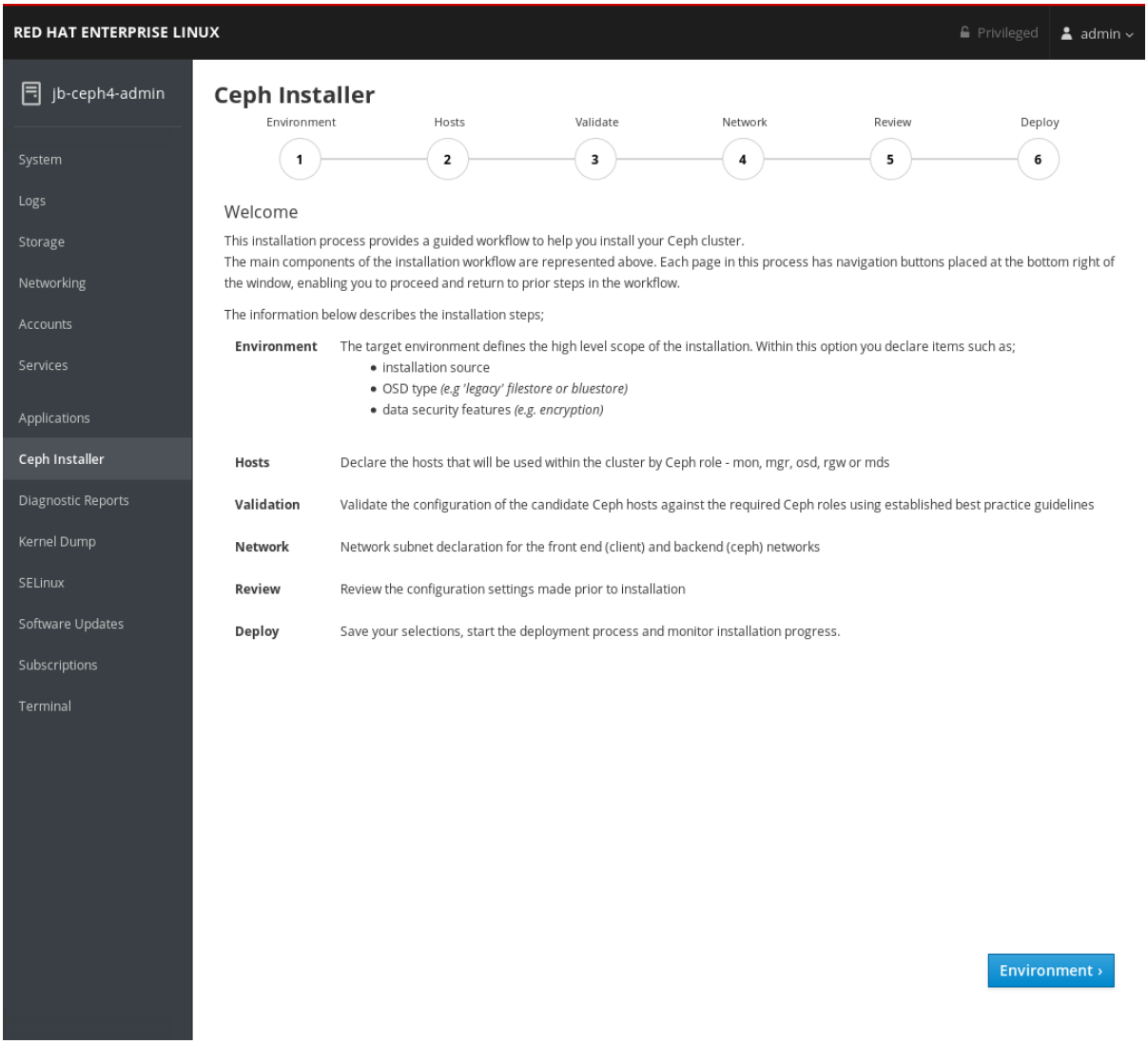
3. 单击 *Reuse my password for privileged tasks* 的单选按钮。



4. 点 *Log In*。



5. 查看欢迎页面以了解安装程序的工作原理和安装过程的整体流程。



在检查了欢迎页面中的信息后，单击 Web 页面右下角的 *环境*按钮。

4.6. 完成 COCKPIT CEPH INSTALLER 的 ENVIRONMENT 页面

Environment 页面允许您配置集群的整体方面，如要使用的安装源，以及如何使用硬盘驱动器 (HDD) 和 Solid State Drives (SSD) 用于存储。

先决条件

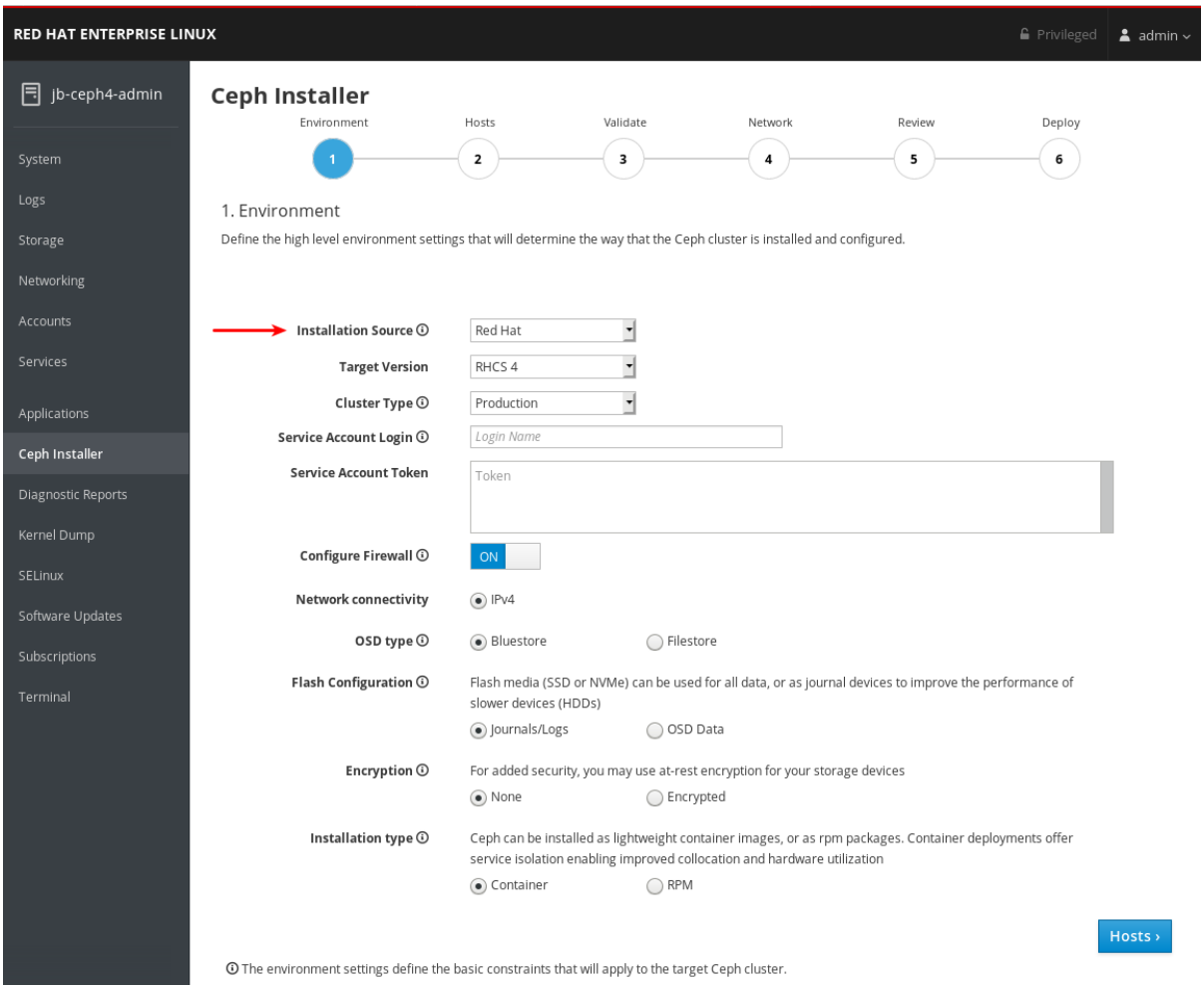
- 已安装并配置了 Cockpit Ceph 安装程序。
- 您有作为配置 Cockpit Ceph 安装程序的一部分输出的 URL。
- 您已创建了 registry 服务帐户。

注意

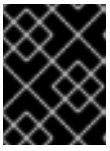
在接下来的对话框中，部分设置右侧有工具提示。若要查看它们，请将鼠标光标悬停在图标上，图标上看起来像一个有圆圈的 *i*。

流程

1. 选择 *Installation Source*。选择 *Red Hat* 以使用来自 Subscription Manager 的存储库，或 ISO 来使用从红帽客户门户下载的 CD 镜像。



如果选择 *Red Hat*，*Target Version* 将被设置为 *RHCS 4* 而没有其他选项。如果选择 *ISO*，*Target Version* 将设置为 ISO 镜像文件。



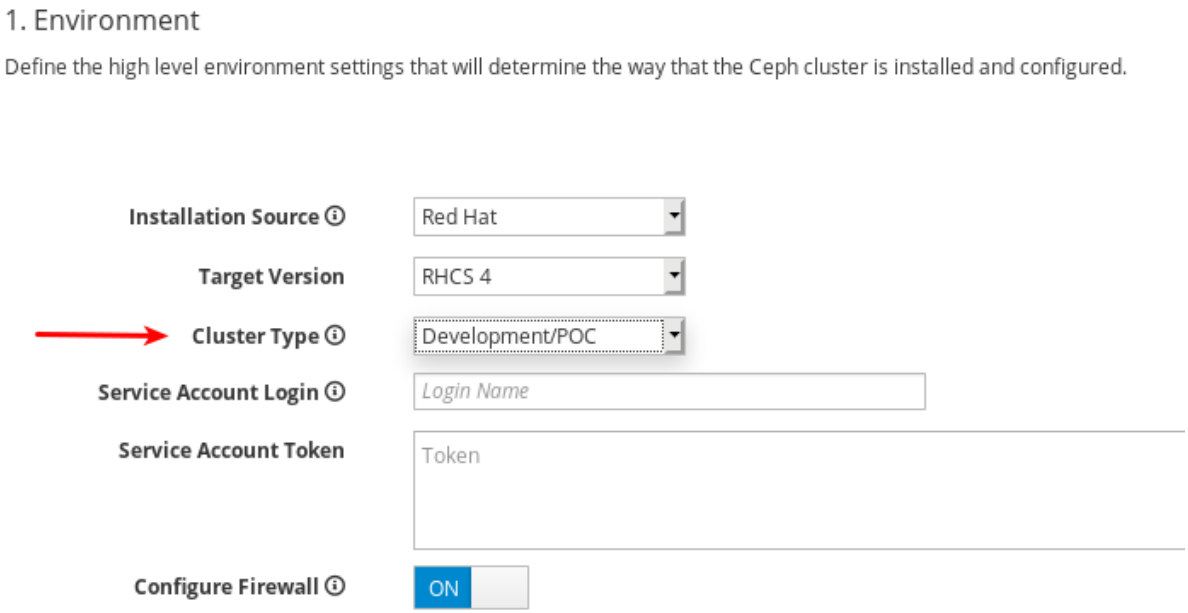
重要

如果选择 ISO，镜像文件必须位于 **/usr/share/ansible-runner-service/iso** 目录中，其 SELinux 上下文必须设置为 **container_file_t**。

重要

Installation Source 的 *Community* 和 *Distribution* 选项不被支持。

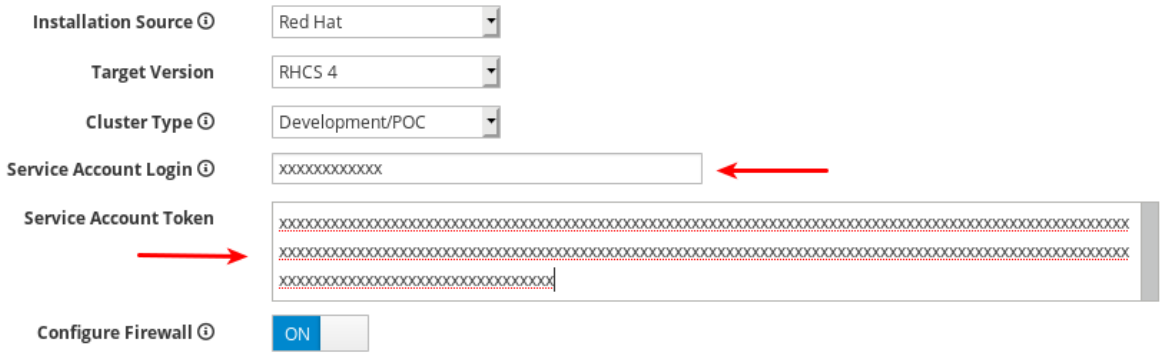
2. 选择 *Cluster Type*。如果未满足 CPU 数量和内存大小等特定资源要求，则 *Production* 选择将阻止安装继续。要允许集群安装即使没有满足资源要求也可以进行，请选择 *Development/POC*。



重要

不要使用 *Development/POC* 模式来安装生产环境中使用的 Ceph 集群。

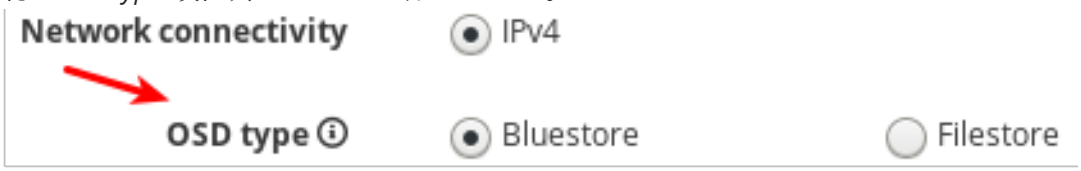
3. 设置*服务帐户登录*和*服务帐户令牌*。如果您没有 Red Hat Registry Service Account，请使用 Registry Service Account 网页 创建一个。



4. 将 *Configure Firewall* 设置为 *ON*，将规则应用到 **firewalld** 以打开 Ceph 服务的端口。如果没有使用 **firewalld**，请使用 *OFF* 设置。

5. 目前，Cockpit Ceph 安装程序仅支持 IPv4。如果您需要 IPv6 支持，个要继续使用 Cockpit Ceph 安装程序，并直接使用 Ansible 脚本 安装 Ceph。

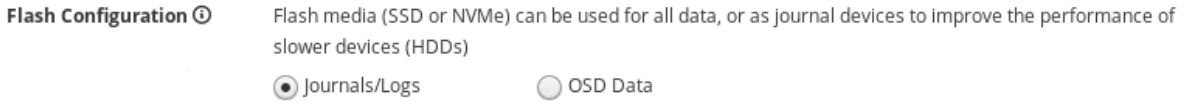
6. 将 *OSD Type* 设置为 *BlueStore* 或 *FileStore*。



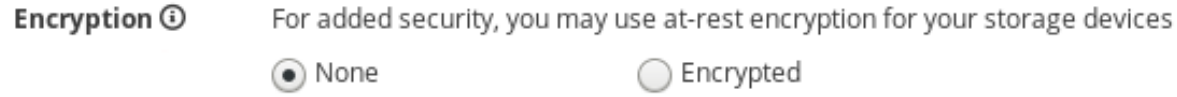
重要

BlueStore 是默认的 OSD 类型。在以前的版本中，Ceph 使用 FileStore 作为对象存储。这种格式对于新的 Red Hat Ceph Storage 4.0 安装已弃用，因为 BlueStore 提供了更多功能和更高的性能。仍可使用 FileStore，但使用它需要支持例外。如需有关 BlueStore 的更多信息，请参阅架构指南 中的 Ceph BlueStore。

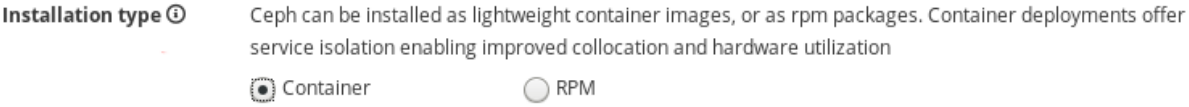
7. 将 *Flash Configuration* 设置为 *Journal/Logs* 或 *OSD 数据*。如果您有 Solid State Drives (SSD)，无论是使用 NVMe 还是传统的 SATA/SAS 接口，您可以选择仅将它们用于写入日志，而实际数据的操作在硬盘驱动器 (HDD) 上进行，或者您可以将 SSD 用于日志和数据，HDD 不用于任何 Ceph OSD 功能。



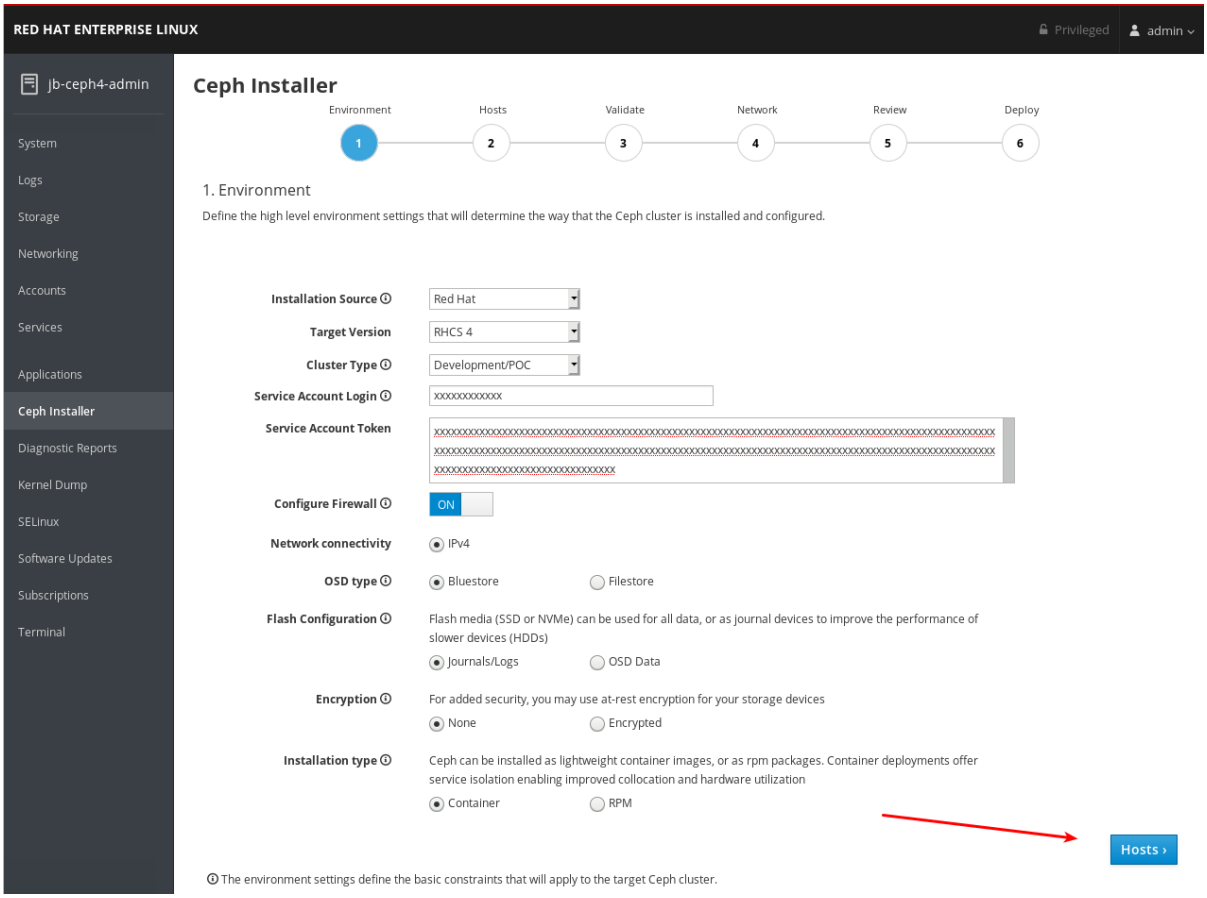
8. 将 *Encryption* 设置为 *None* 或 *Encrypted*。这是指使用 LUKS1 格式的存储设备的其余加密。



9. 将 *Installation 类型* 设置为 *Container* 或 *RPM*。传统上，红帽软件包管理器 (RPM) 用于在 Red Hat Enterprise Linux 上安装软件。现在，您可以使用 RPM 或容器安装 Ceph。使用容器安装 Ceph 可以提高硬件利用率，因为可以隔离和并置服务。



10. 查看所有环境设置，再单击网页右下角的 *Hosts* 按钮。



4.7. 完成 COCKPIT CEPH INSTALLER 的 HOSTS 页面

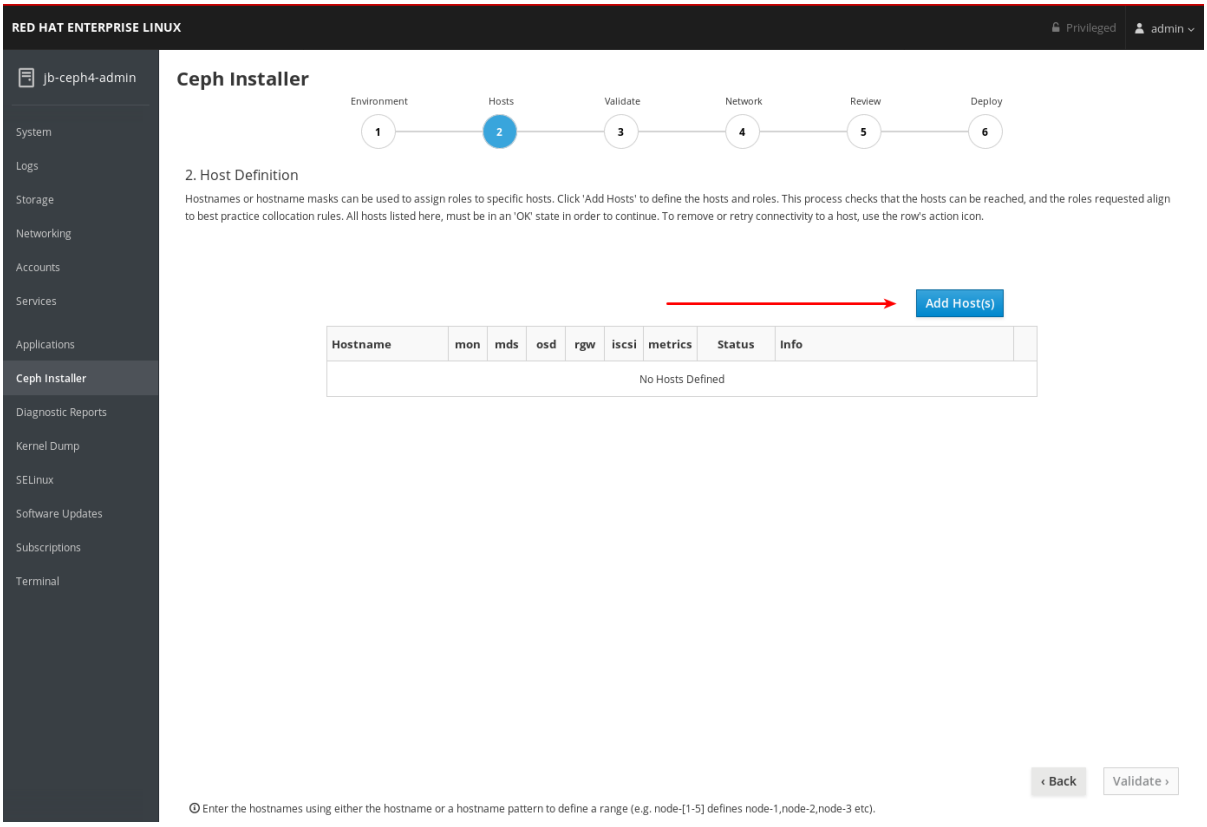
借助 *Hosts* 页面，您可以向 Cockpit Ceph 安装程序通知要在其上安装 Ceph 的主机，以及每个主机将用作的角色。当您添加主机时，安装程序将检查它们是否有 SSH 和 DNS 连接。

先决条件

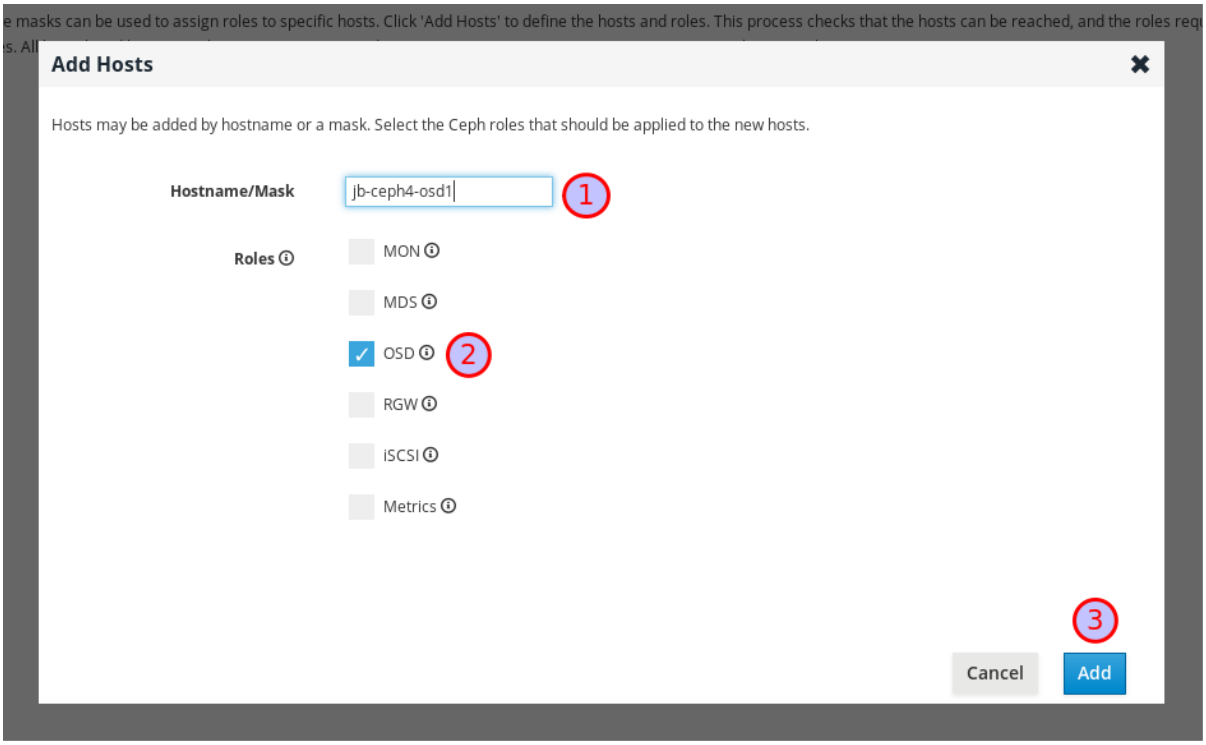
- Cockpit Ceph 安装程序的环境页面已完成。
- Cockpit Ceph Installer SSH 密钥已复制到集群中的所有节点。

流程

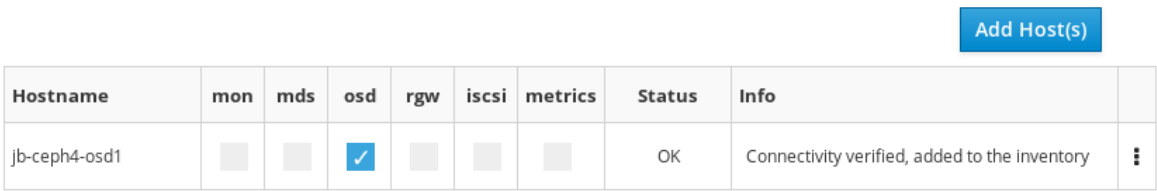
1. 单击 *添加主机按钮*。



2. 输入 Ceph OSD 节点的主机名，选中 *OSD* 的方框，然后单击 *Add* 按钮。

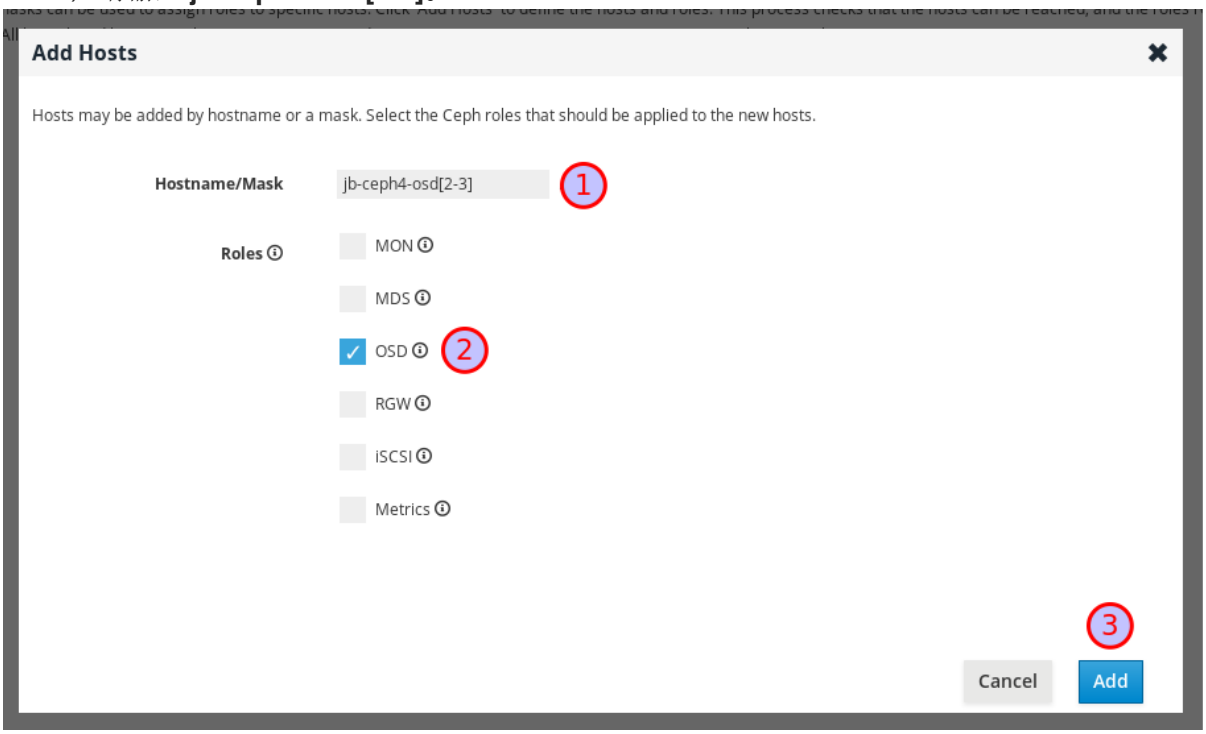


添加了第一个 Ceph OSD 节点。

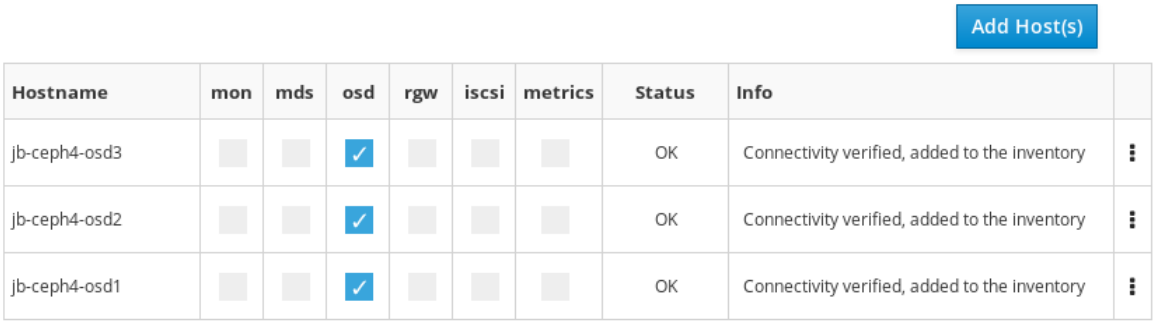


对于生产环境集群，请重复此步骤，直到您至少添加了三个 Ceph OSD 节点。

3. 可选：使用主机名模式来定义节点范围。例如，若要同时添加 **jb-ceph4-osd2** 和 **jb-ceph4-osd3**，请输入 **jb-ceph4-osd[2-3]**。



添加 **jb-ceph4-osd2** 和 **jb-ceph4-ods3**。



4. 为集群中的其他节点重复上述步骤。

   a. 对于生产环境集群，至少添加三个 Ceph 监控节点。在对话框中，该角色列为 **MON**。

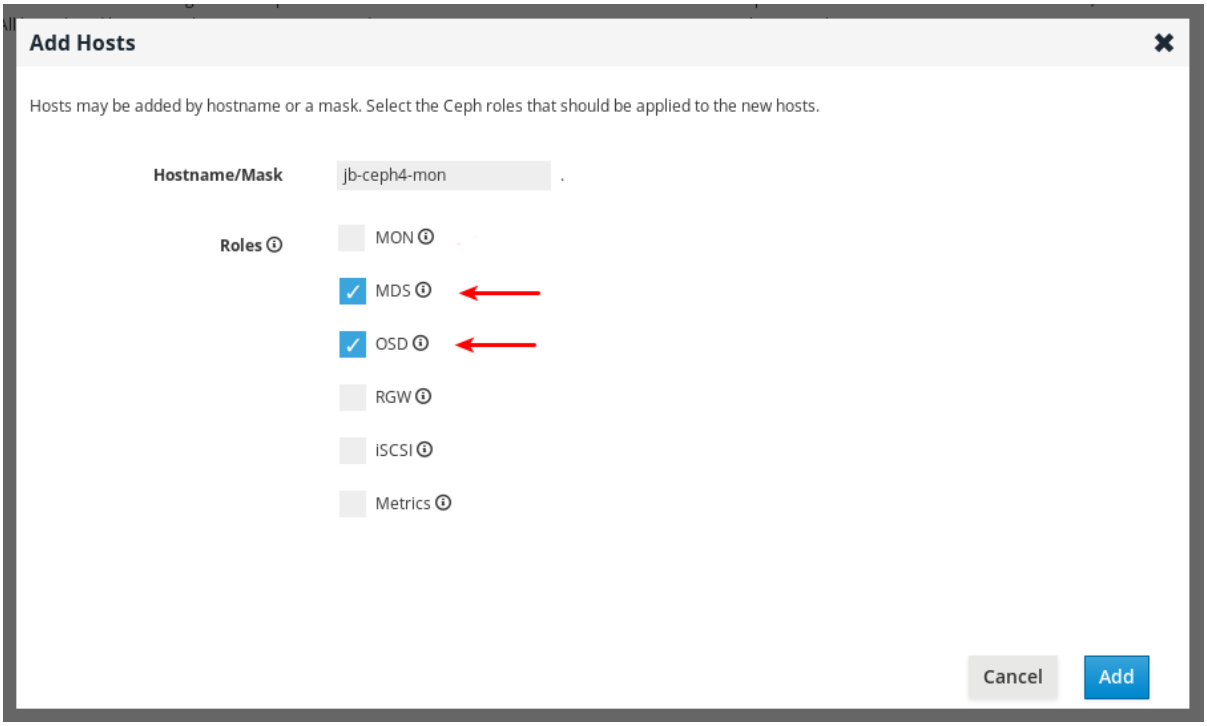
   b. 使用 **Metrics** 角色添加节点。**Metrics** 角色安装 Grafana 和 Prometheus，以提供 Ceph 集群性能的实时洞察。这些指标显示在 Ceph 控制面板中，您可以通过它来监控和管理集群。需要安装仪表板、Grafana 和 Prometheus。您可以在 Ansible 管理节点上并置指标功能。如果这样做，请确保节点的系统资源大于独立指标节点所需的值。

   c. 可选：添加具有 **MDS** 角色的节点。**MDS** 角色安装 Ceph 元数据服务器 (MDS)。元数据服务器守护进程是部署 Ceph 文件系统所必需的。

   d. 可选：使用 **RGW** 角色添加节点。**RGW** 角色安装 Ceph 对象网关，也称为 RADOS 网关，这是在 librados API 基础上构建的对象存储接口，为应用提供 Ceph 存储集群的 RESTful 网关。它支持 Amazon S3 和 OpenStack Swift API。

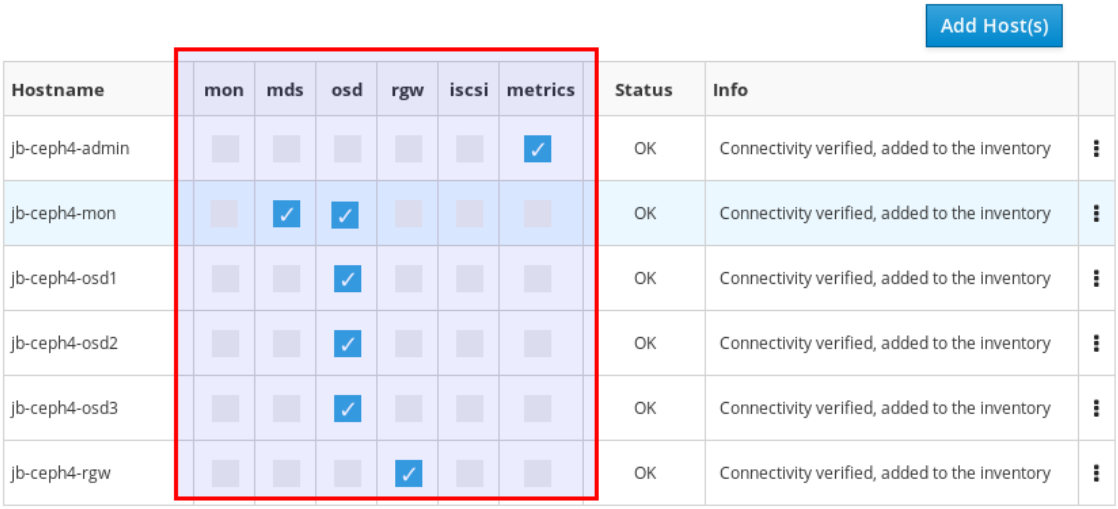
   e. 可选：添加具有 **iSCSI** 角色的节点。**iSCSI** 角色安装 iSCSI 网关，以便您可以通过 iSCSI 共享 Ceph 块设备。要将 iSCSI 与 Ceph 搭配使用，您必须在至少两个用于多路径 I/O 的节点上安装 iSCSI 网关。

5. 可选：在添加节点时通过选择多个角色在同一节点上并分配多个服务。

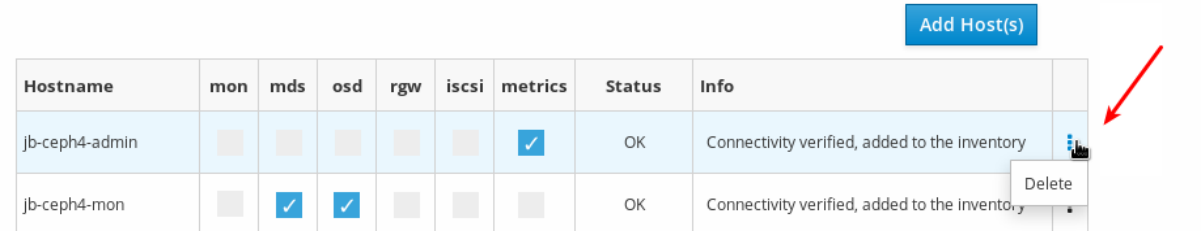


如需有关共同定位守护进程的更多信息，请参阅安装指南中的容器化 Ceph 守护进程的重新定位。

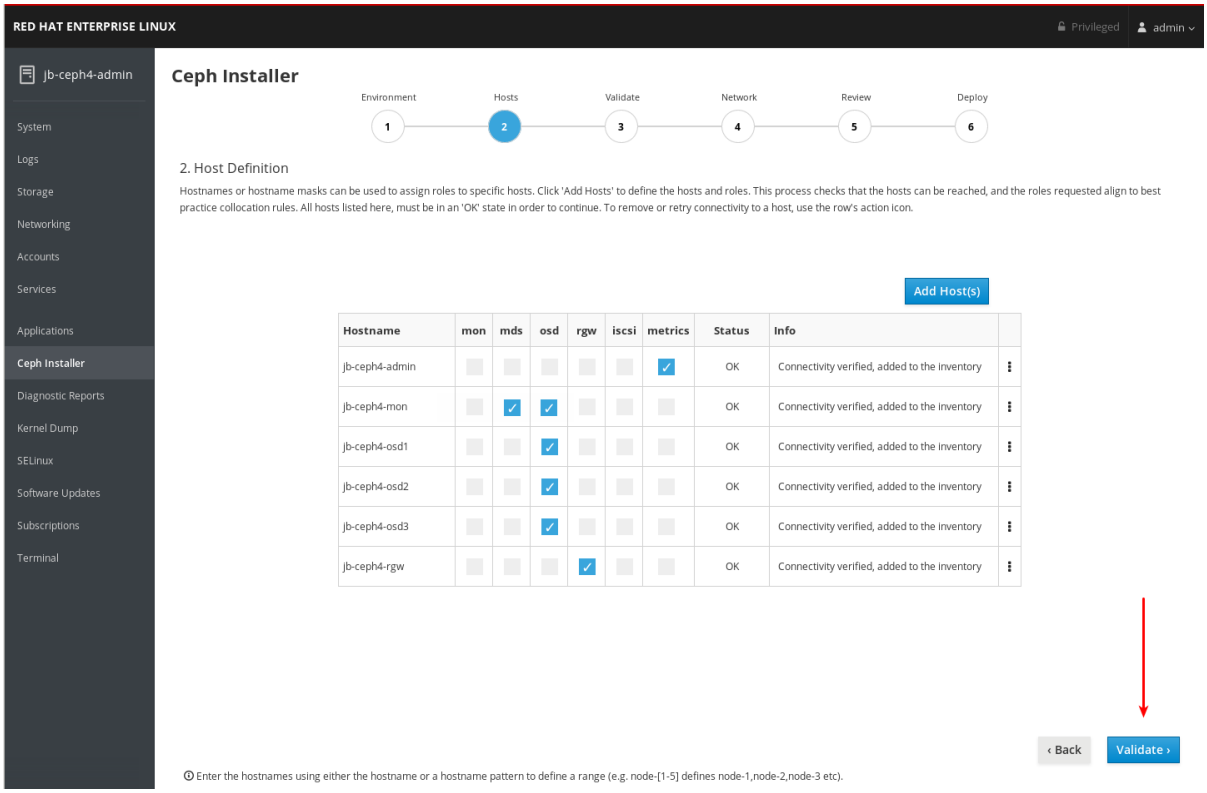
6. 可选：通过检查或取消检查表中的角色来修改分配给节点的角色。

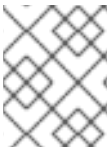


7. 可选： 要删除节点，在您要删除的节点所在行的右侧点击 kebab 图标，然后点 *Delete*。



8. 添加集群中的所有节点并设置所有必要的角色后，单击页面右下角的 *Validate* 按钮。





注意

对于生产集群，除非有三个或五个 monitor，否则 Cockpit Ceph 安装程序将无法继续。在这些示例中，*Cluster Type* 设置为 *Development/POC*，因此安装只能进行一个 monitor。

4.8. 完成 COCKPIT CEPH INSTALLER 的 VALIDATE 页面

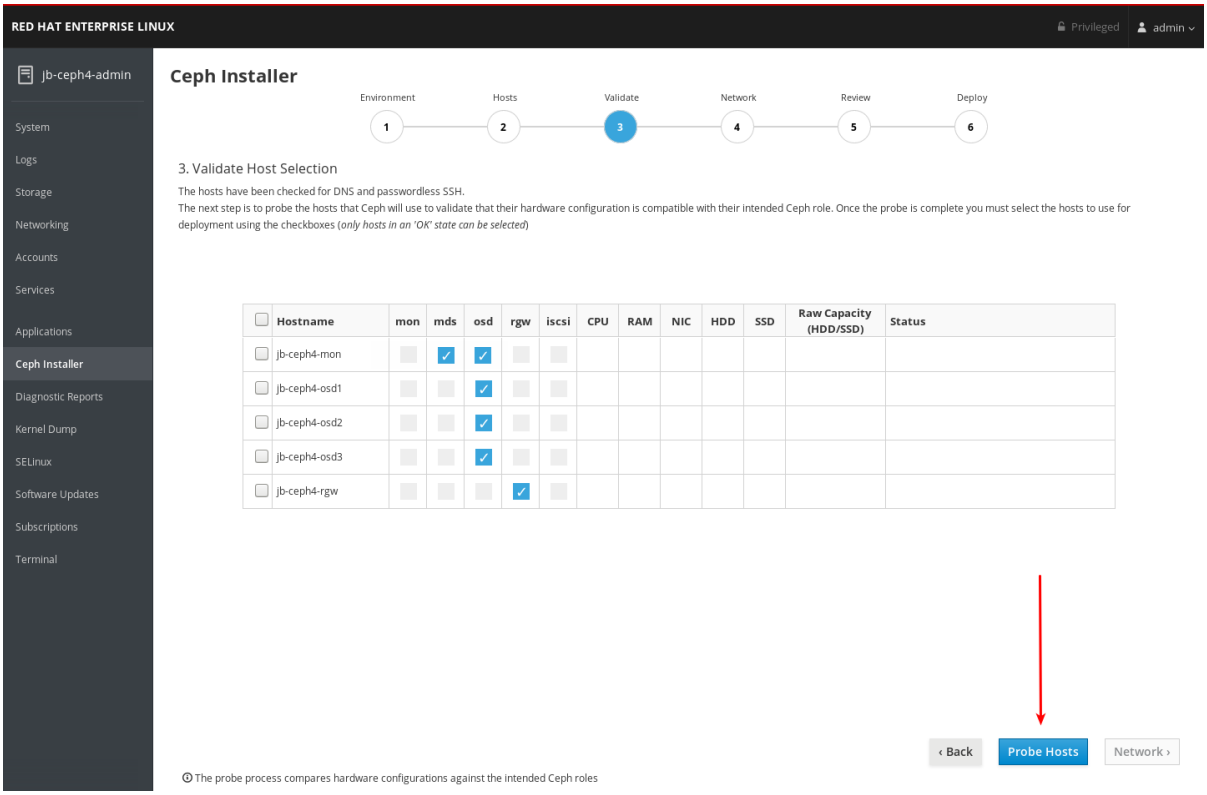
通过 *Validate* 页面，您可以探测到 *Hosts* 页面上提供的节点，以验证它们是否满足您要用于它们的角色的硬件要求。

先决条件

- Cockpit Ceph Installer 的 Hosts 页面 已完成。

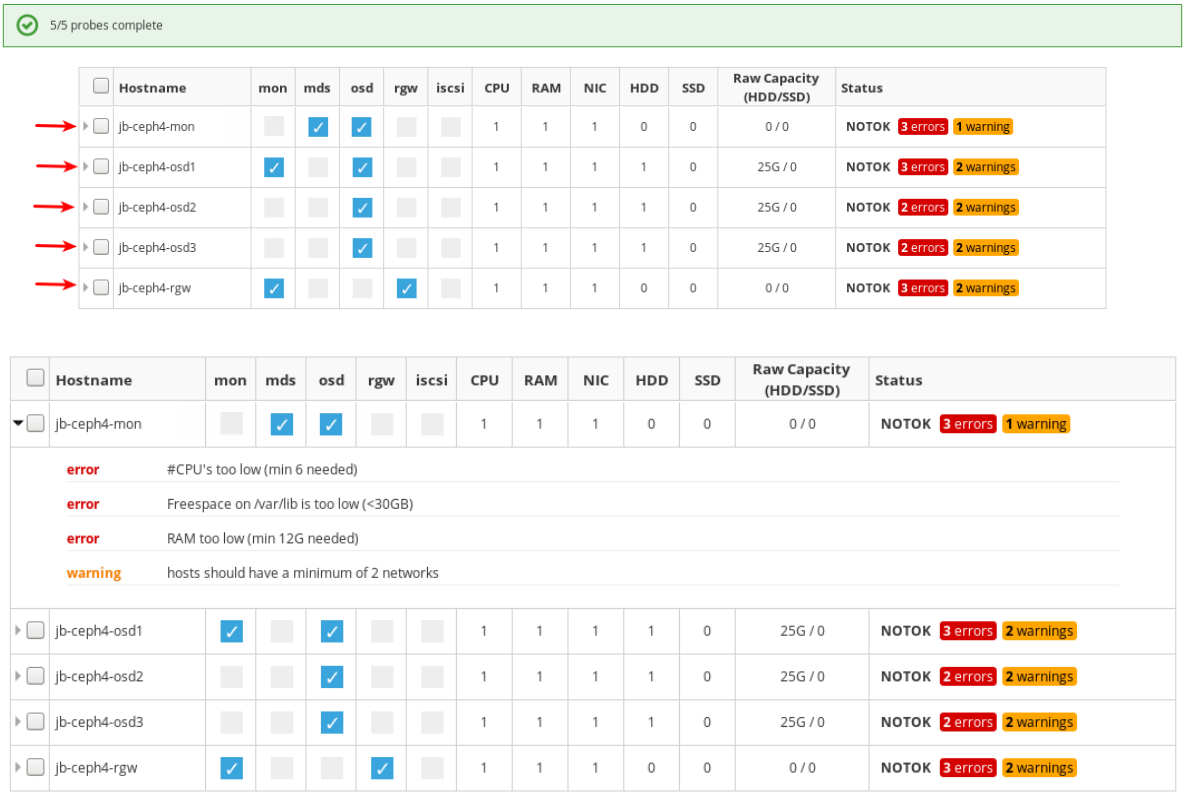
流程

1. 点*探测主机*按钮。



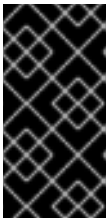
要继续，您必须至少选择三个具有 *OK 状态*的主机。

2. 可选：如果为主机生成了警告或错误，请点击主机检查标记左侧的箭头来查看问题。



重要

如果将 *Cluster Type* 设置为 *Production*，则生成的任何错误都将导致 *Status* 变为 *NOTOK*，您将无法选择它们进行安装。阅读下一步以了解有关如何解决错误的信息。



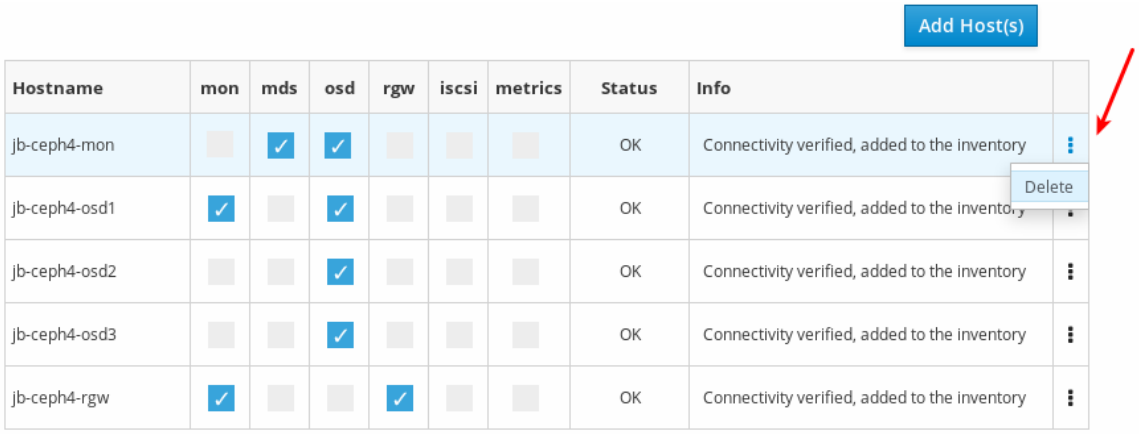
重要

如果将 *Cluster Type* 设为 *Development/POC*，则生成的任何错误都将列为警告，因此 *Status* 始终为 *OK*。这样，您可以选择主机并在主机上安装 Ceph，无论主机是否满足要求或建议。如果您想解决警告，您仍然可以解决。阅读下一步以了解有关如何解析警告的信息。

3. 可选：要解决错误和警告，请使用以下一个或多个方法。

   a. 解决错误或警告的最简单方法是完全禁用某些角色，或者在一个主机上禁用角色并在具有所需资源的另一主机上启用它。
   对启用或禁用角色进行完全的测试，直到找到适当的组合，如果您安装 *Development/POC* 集群，您就可以继续处理任何剩余的警告，或者如果您安装*生产*集群，至少三个主机具有分配给它们的角色所需的资源，并且您能够轻松处理任何剩余的警告。

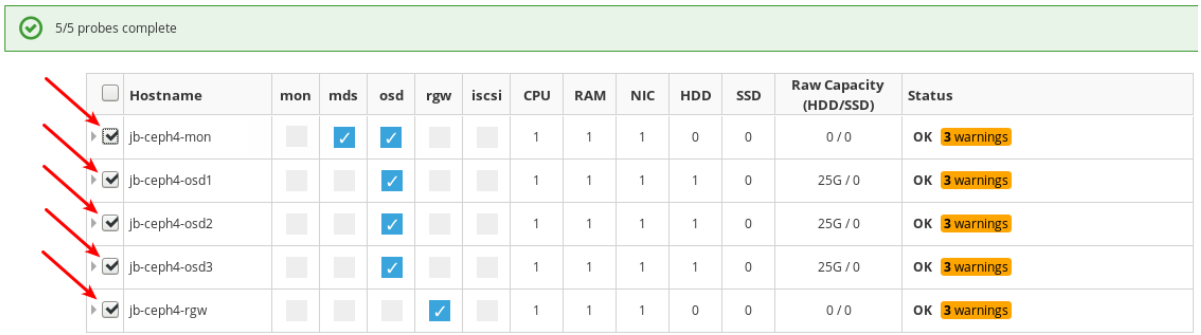
   b. 您还可以使用满足所需角色要求的新主机。首先返回到 *Hosts* 页面，再删除有问题的主机。



   然后，添加新主机。

   c. 如果您要在主机上升级硬件或以某种其他方式对其进行修改，这样它将满足要求或建议，首先对主机进行所需的更改，然后再次点*探测主机*。如果必须重新安装操作系统，您必须再次复制 SSH 密钥。

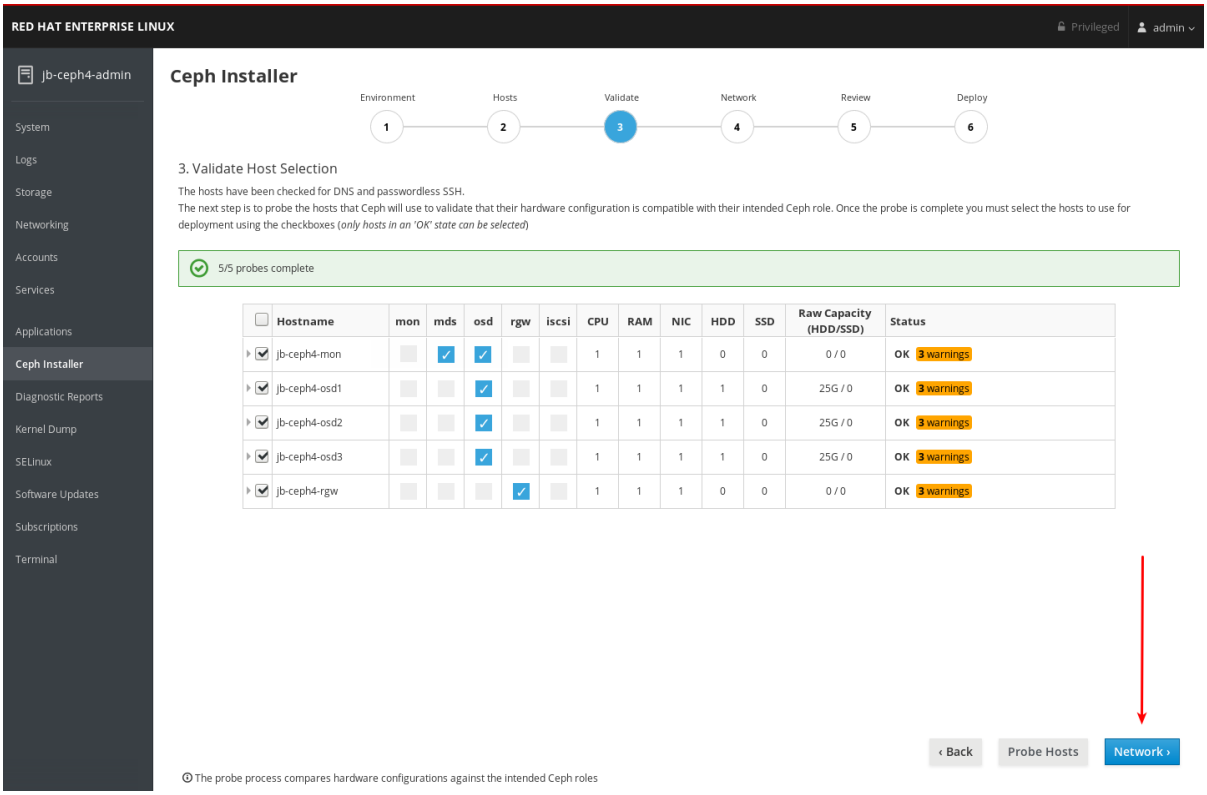
4. 通过选中主机旁边的框，选择要在其上安装 Red Hat Ceph Storage 的主机。



重要

如果安装生产环境集群，您必须先解决所有错误，然后才能选择它们进行安装。

5. 单击页面右下角的 *Network* 按钮，以查看并配置集群的网络。



4.9. 完成 COCKPIT CEPH INSTALLER 的 NETWORK 页面

Network 页面允许您隔离某些集群通信类型到特定网络。这需要在集群中的主机之间配置多个不同的网络。

重要

Network 页面使用从 *Validate* 页面执行的探测中收集的信息来显示您的主机可以访问的网络。目前，如果您已进入 *Network* 页面，则无法向主机添加新网络，返回到 *Validate* 页面，重新检测主机，然后再次继续 *Network* 页面并使用新的网络。它们将不会显示为选择。若要在已进入 *Network* 页面后使用添加到主机的网络，您必须完全刷新 Web 页面，然后从开始重新开始安装。

重要

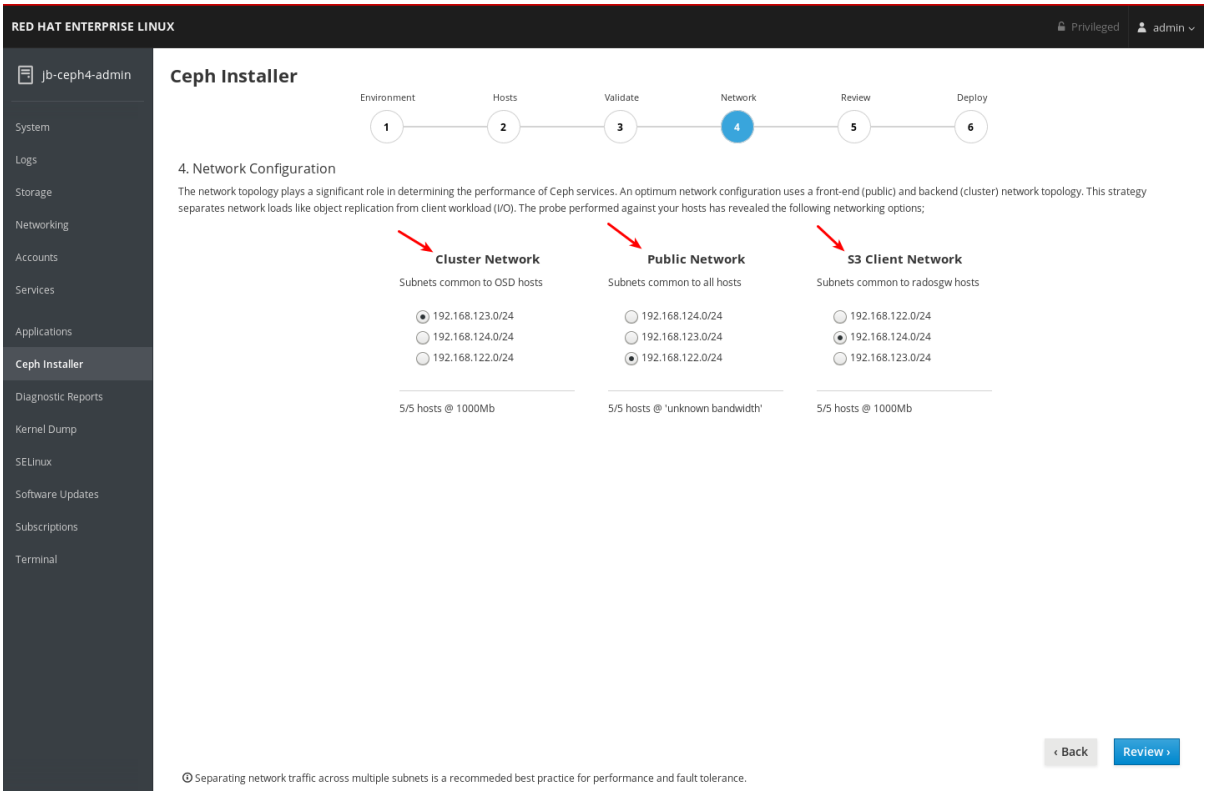
对于生产环境集群，您必须将集群内流量与独立 NIC 上的 client-to-cluster 流量隔离。除了隔离集群流量类型外，设置 Ceph 集群时需要考虑其他网络注意事项。如需更多信息，请参阅硬件指南中的网络注意事项。

先决条件

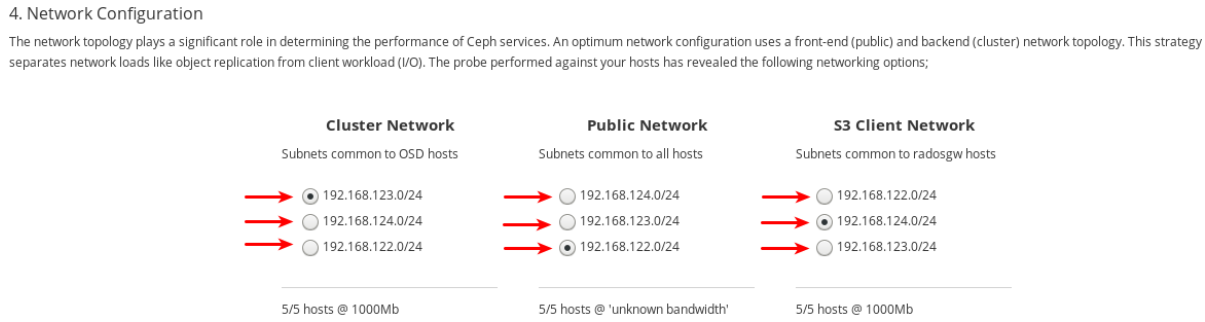
- Cockpit Ceph Installer 的 Validate 页面 已完成。

流程

1. 记录您可以在 Network 页面中配置的网络类型。每种类型都有自己的列。始终显示 *Cluster Network* 和 *Public Network* 的列。如果您使用 RADOS 网关角色安装主机，则将显示 *S3 Network* 列。如果您使用 iSCSI 角色安装主机，则将显示 *iSCSI Network* 列。在以下示例中，显示了 *Cluster Network*、*Public Network* 和 *S3 Network* 的列。

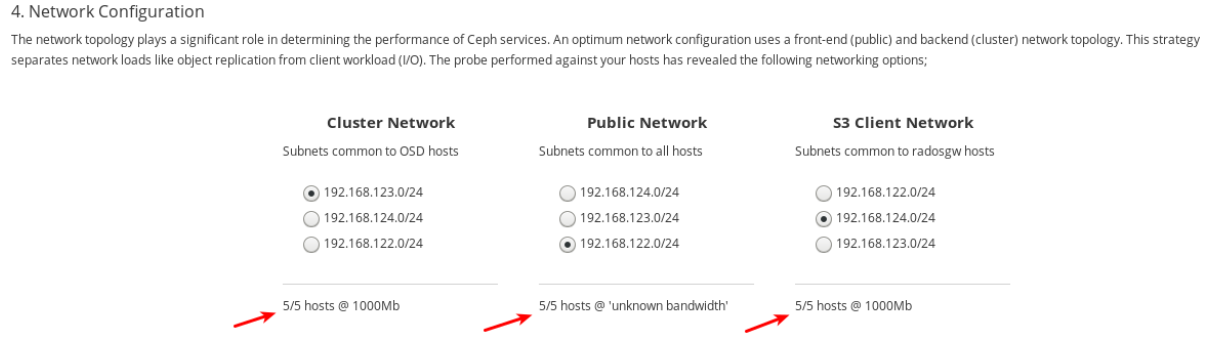


2. 记录您可以为每种网络类型选择的网络。仅显示所有组成特定网络类型的主机上可用的网络。在以下示例中，集群中的所有主机上都提供了三个网络。由于所有三个网络都在构成网络类型的每组主机上都可用，每种网络类型列出了相同的三个网络。

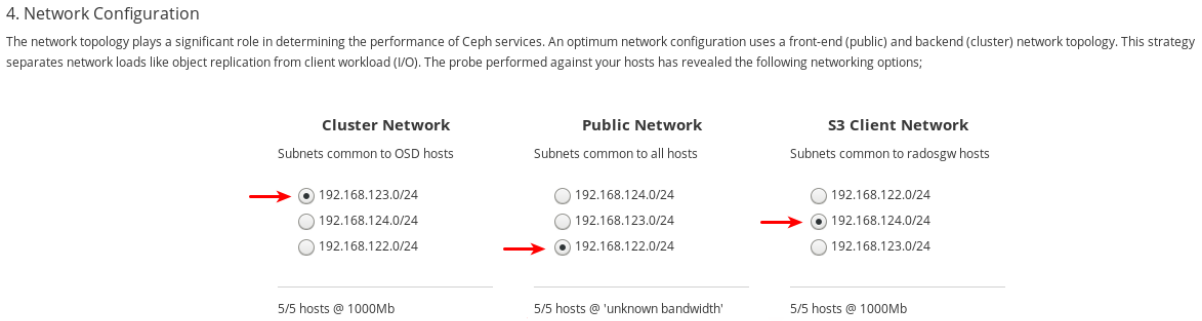


可用的三个网络 为 **192.168.122.0/24**、192**.168.123.0/24** 和 **192.168.124.0/24**。

3. 记录每个网络在运行的速度。这是用于特定网络的 NIC 的速度。在以下示例中，**192.168.123.0/24** 和 **192.168.124.0/24** 为 1,000 mbps。Cockpit Ceph 安装程序无法确定 **192.168.122.0/24** 网络的速度。

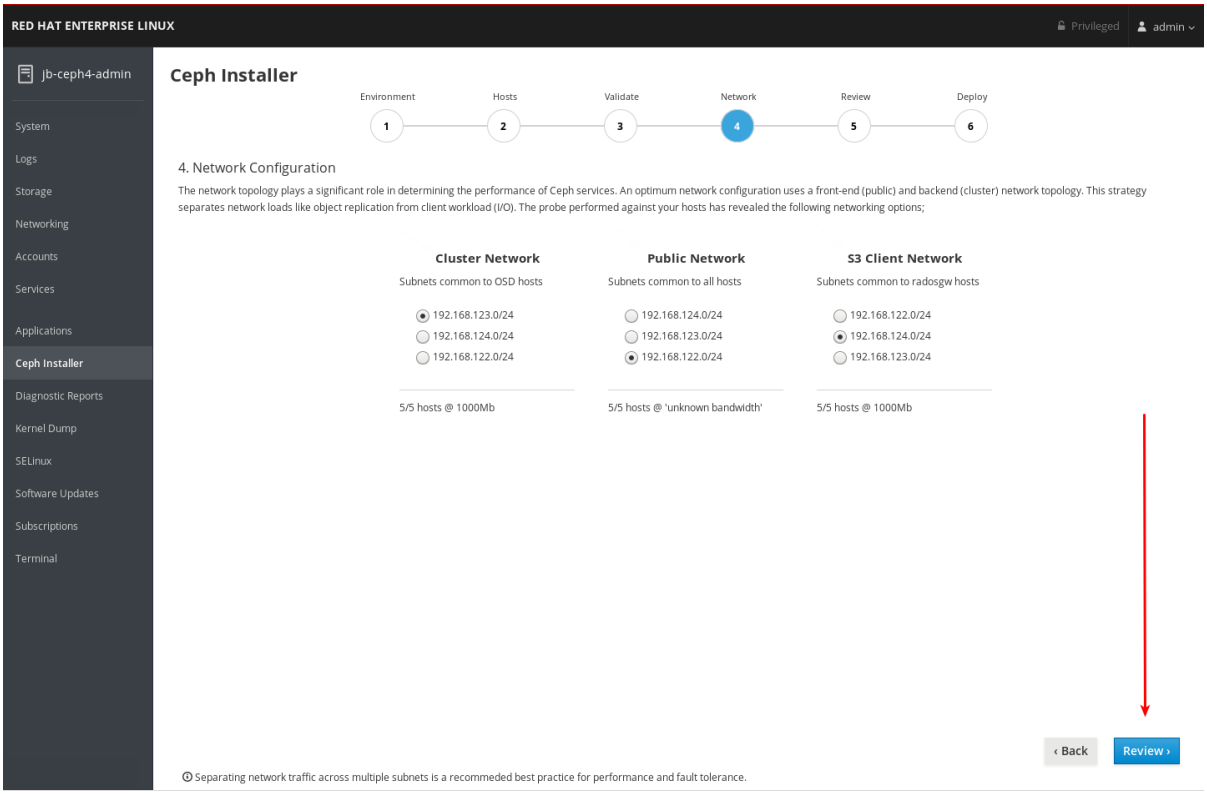


4. 选择您要用于每种网络类型的网络。对于生产环境集群，您必须为 *Cluster Network 和 Public Network* 选择单独的网络。对于开发/POC 集群，可以为这两种类型选择相同的网络，或者如果所有主机上都只有一个网络，则只会显示该网络，您将无法选择其他网络。



192.168.122.0/24 网络将用于 *Public Network*，**192.168.123.0/24** 网络将用于 *Cluster Network*，**192.168.124.0/24** 网络将用于 *S3 Network*。

5. 在安装前，点页面右下角的 *Review* 按钮来查看整个集群配置。



4.10. 查看安装配置

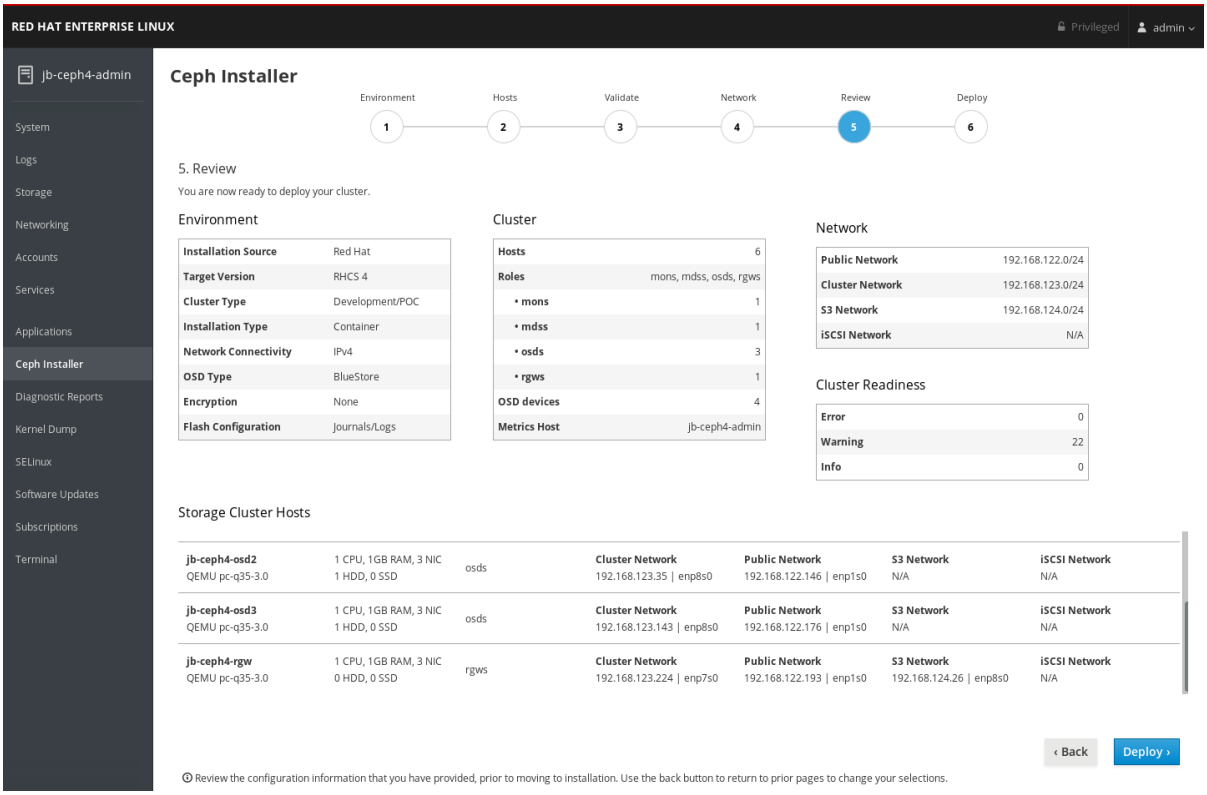
借助 *Review* 页面，您可以查看在上一页上设置的 Ceph 集群安装配置的所有详情，以及主机的详细信息，其中一些未包含在上一页中。

先决条件

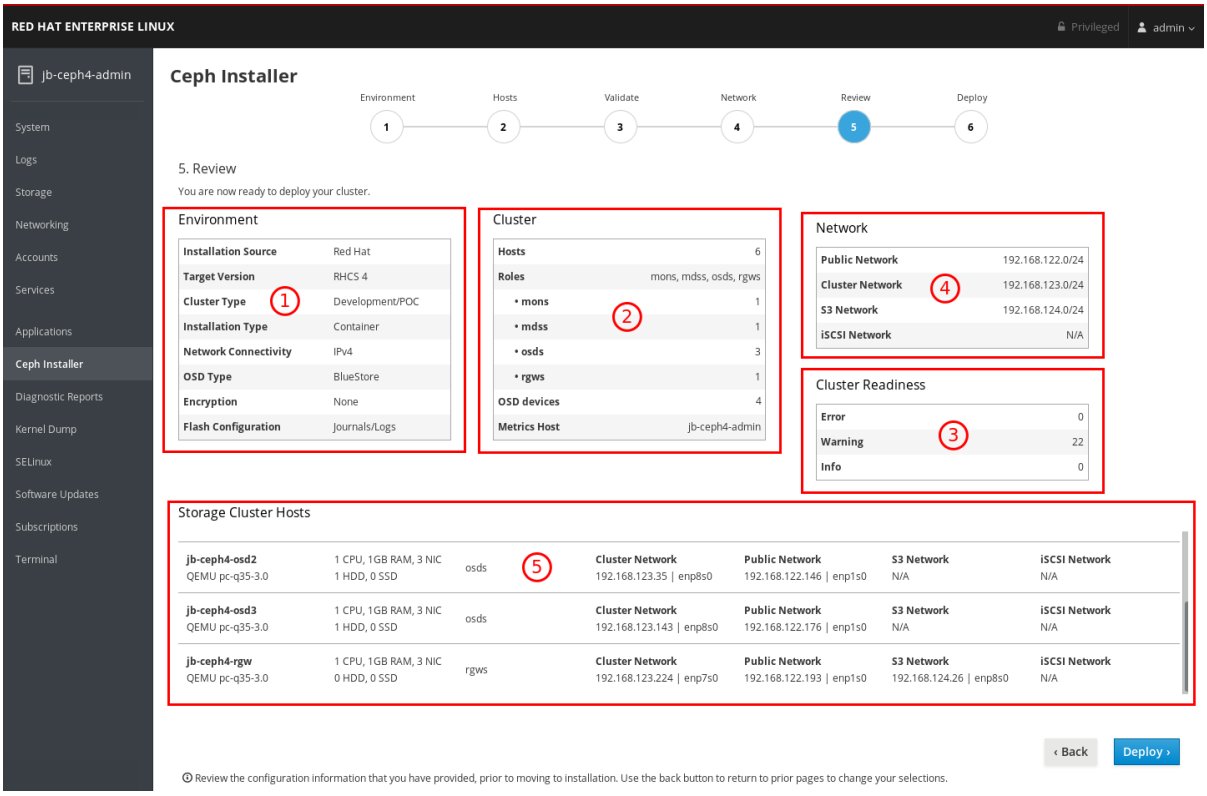
- 完成 Cockpit Ceph 安装程序的 Network 页面。

流程

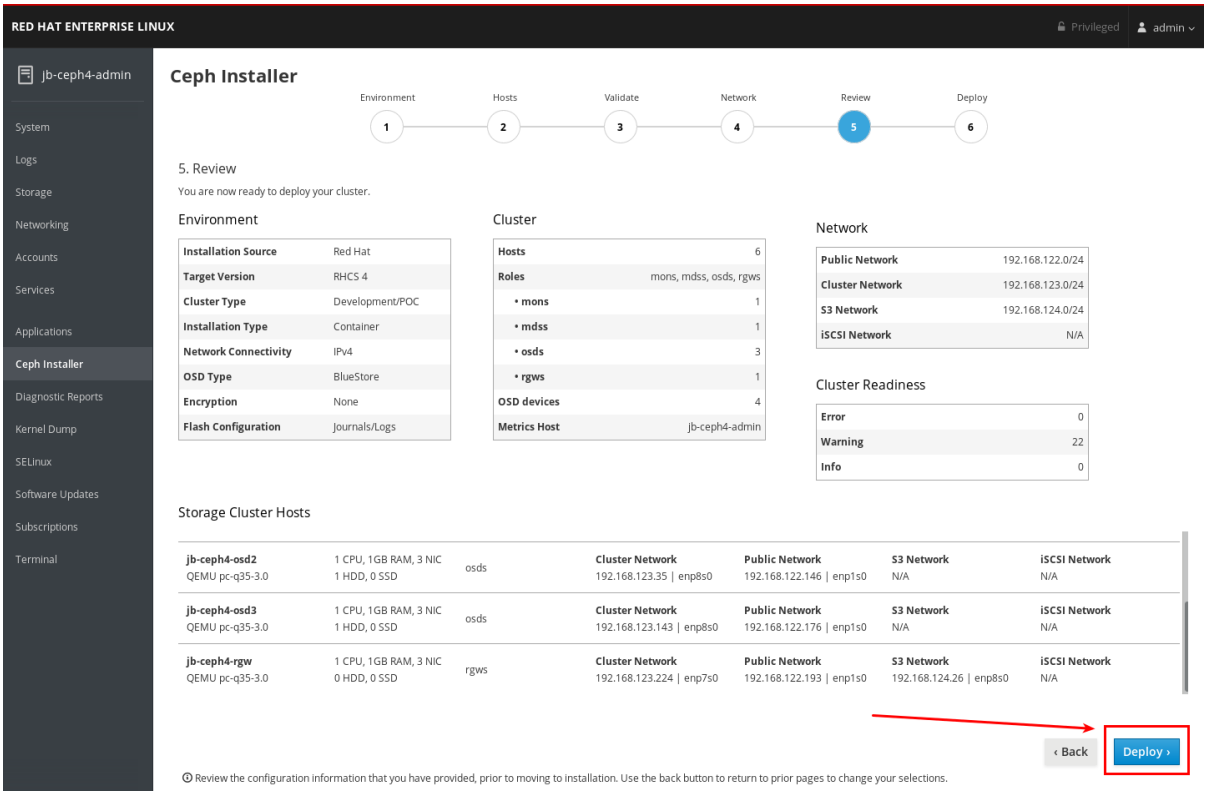
1. 查看检查页面。



2. 验证上一页中的信息如您预期的那样，如 *Review* 页面所示。来自 *Environment* 页面的信息摘要位于 **1**，后跟着 *Hosts* 页面位于 **2**，*Validate* 页面位于 **3**，*Network* 页面位于 **4**，以及主机的详细信息（包括前面页面中未包含的一些额外详细信息），位于 **5**。



3. 单击页面右下角的 *Deploy* 按钮，以进入 *Deploy* 页面，您可以在其中完成并启动实际安装过程。



4.11. 部署 CEPH 集群

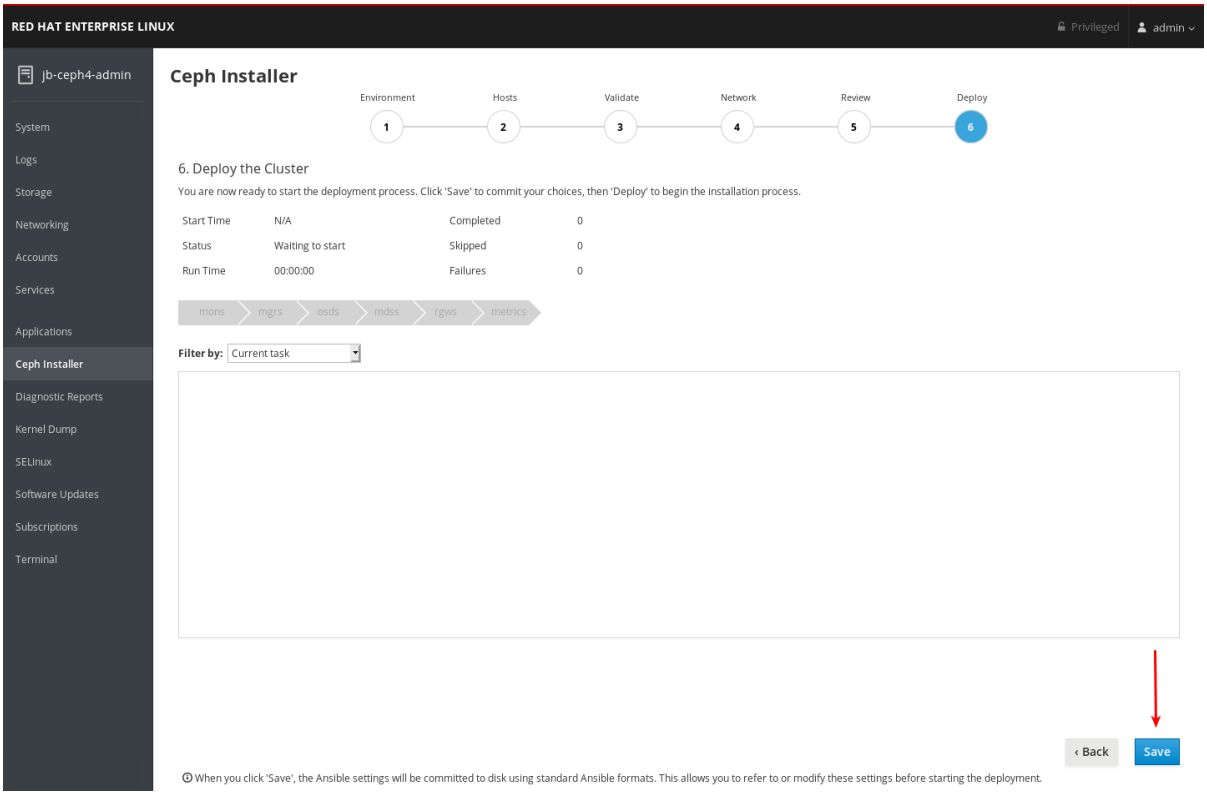
Deploy 页面允许您以原生 Ansible 格式保存安装设置，并在需要时检查或修改它们，启动安装，监控其进度，并在安装成功完成后查看集群的状态。

先决条件

- 检查 Review 页中的安装配置设置。

流程

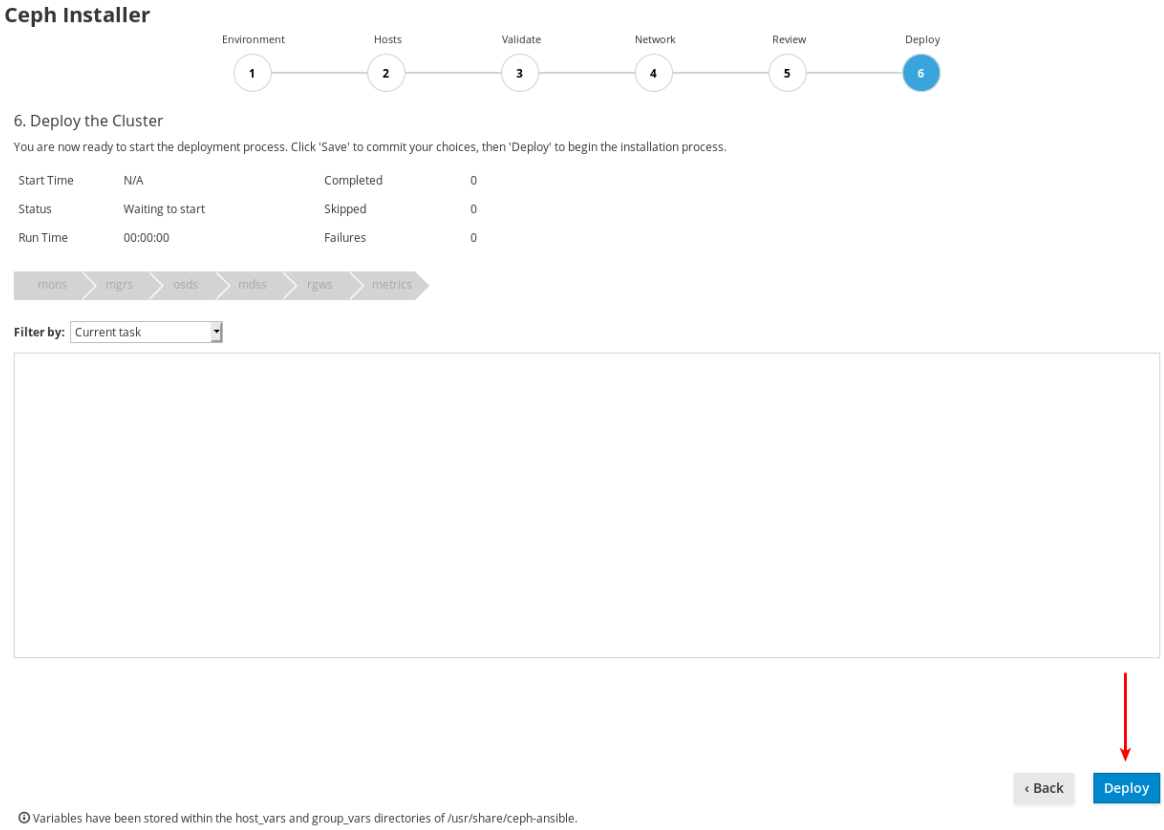
1. 单击页面右下角的 *Save* 按钮，将安装设置保存到 Ansible playbook 中，供 Ansible 用于执行实际安装。



2. 可选：查看或进一步自定义位于 Ansible 管理节点上的 Ansible playbook 中的设置。playbook 位于 **/usr/share/ceph-ansible**。如需有关 Ansible playbook 以及如何使用它们自定义安装的更多信息，请参阅 安装 Red Hat Ceph Storage 集群 。

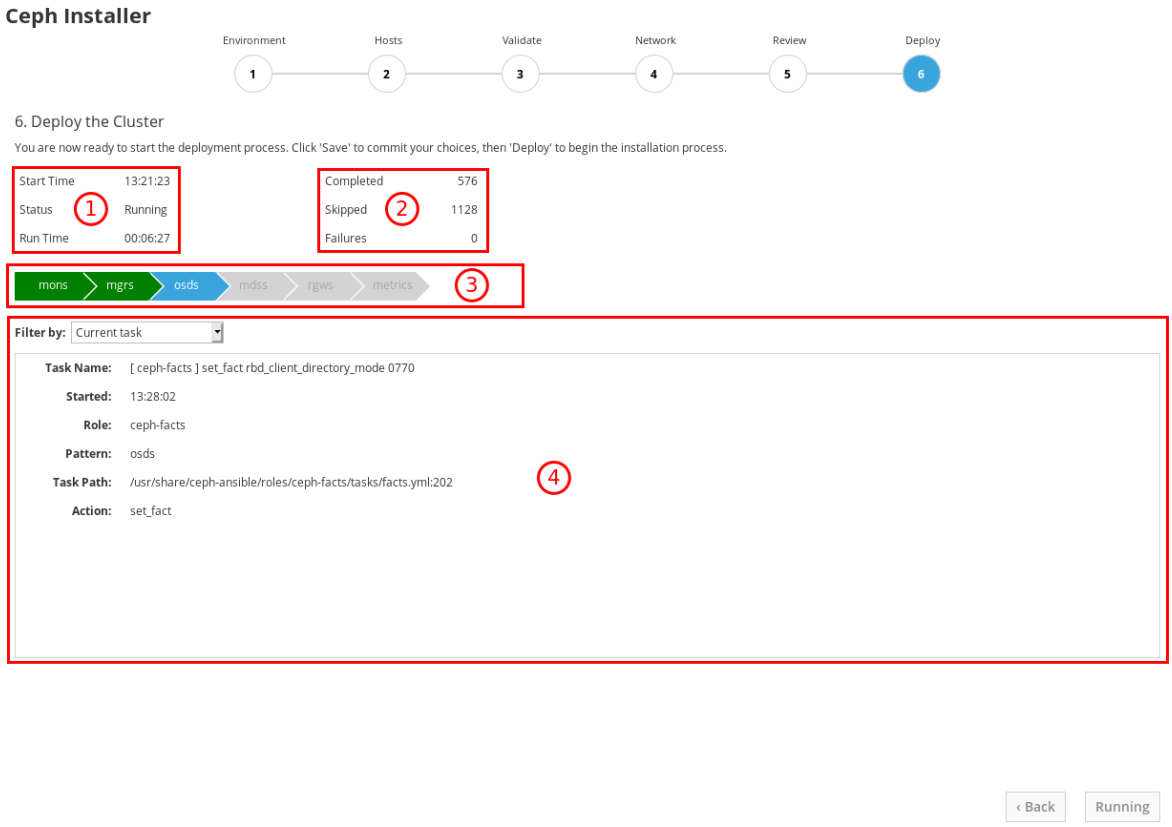
3. 为 Grafana 和仪表板保护默认用户名和密码。从 Red Hat Ceph Storage 4.1 开始，您必须在 **/usr/share/ceph-ansible/group_vars/all.yml** 中取消注释或设置 **dashboard_admin_password** 和 **grafana_admin_password**。为每个用户设置安全密码.另外，为 **dashboard_admin_user** 和 **grafana_admin_user** 设置自定义用户名。

4. 单击页面右下角的 *Deploy* 按钮以开始安装。



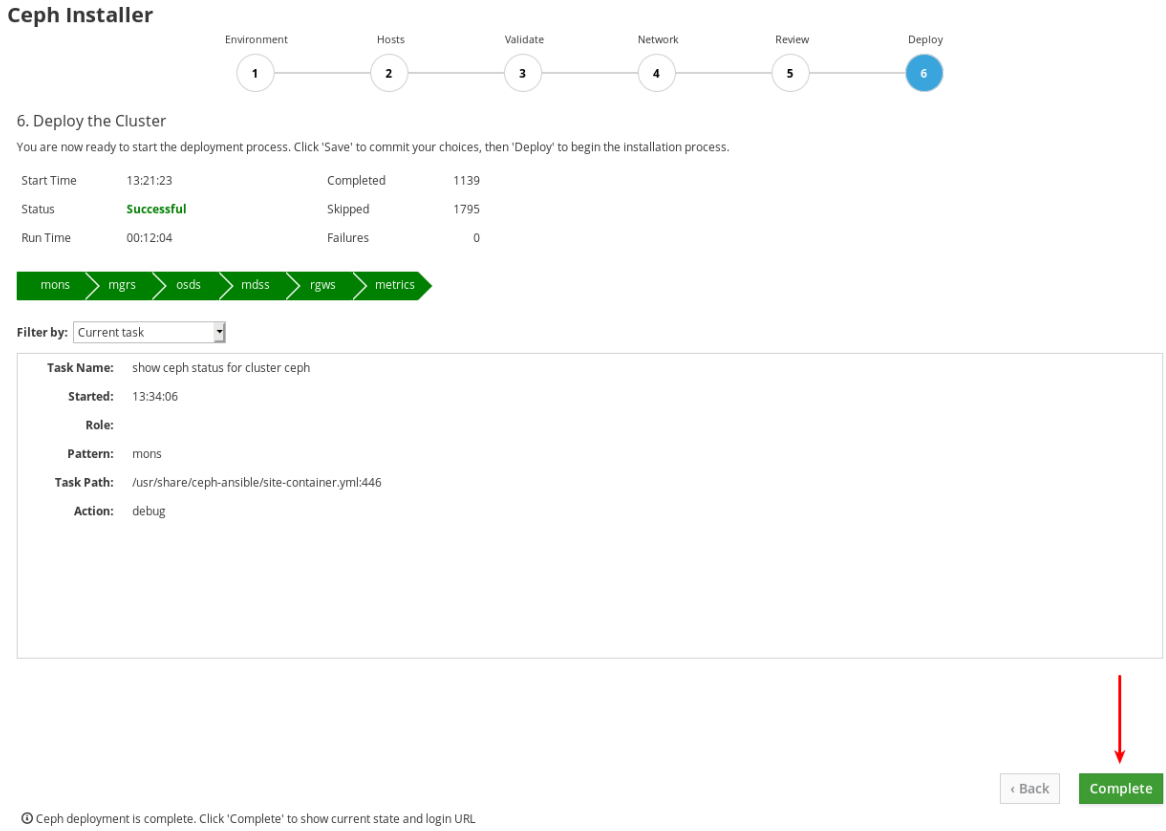
5. 观察正在运行时的安装进度。

   1 处的信息显示安装是否正在运行、开始时间和已过时间。2 的信息显示了已尝试的 Ansible 任务的摘要。3 的信息显示了已安装或正在安装的角色。绿色表示一个角色，其中分配了该角色的所有主机都已安装了该角色。蓝色代表了一个角色，其中分配了该角色的主机仍然被安装。在 4 中，您可以查看当前任务的详情或查看失败的任务。使用 *Filter by* 菜单在当前任务和失败的任务之间切换。



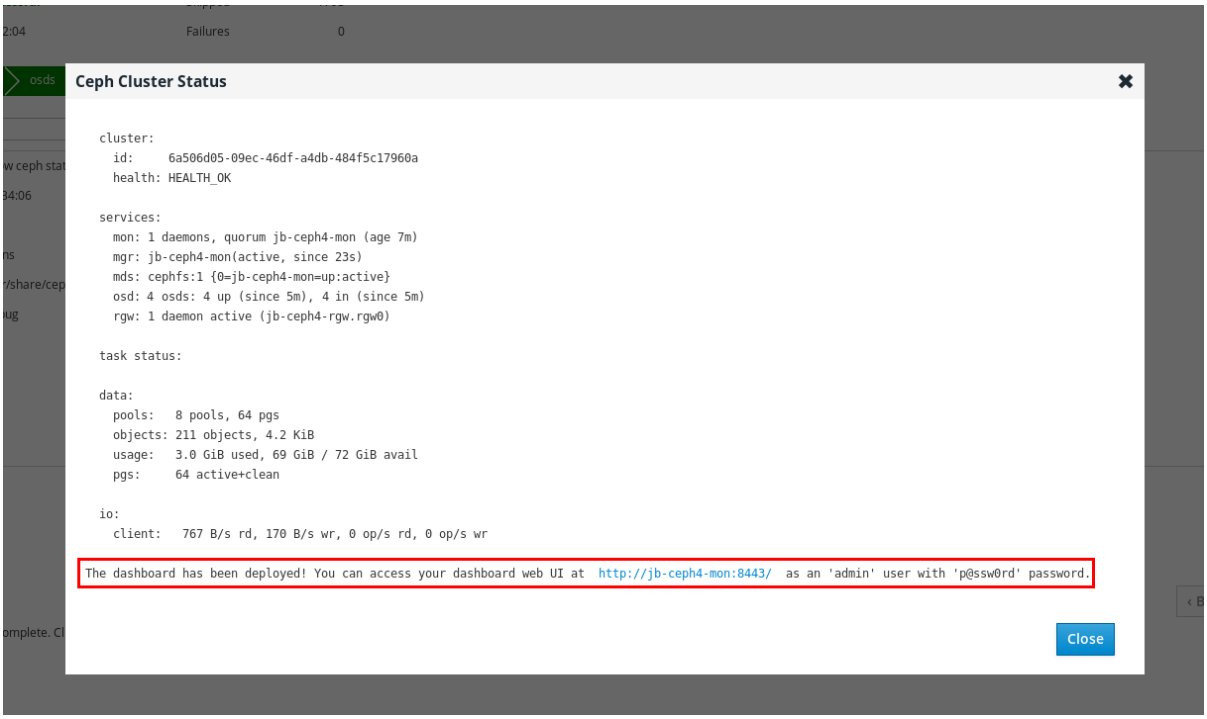
角色名称来自 Ansible 清单文件。这等同于：**mons** 是 Monitor， **mgrs** 是 Manager，请注意 Manager 角色与 Monitor 角色一起安装，**osds** 是对象存储设备，**mdss** 是元数据服务器，**rgws** 是 RADOS 网关，**metrics** 是 Grafana，用于仪表板指标。示例屏幕截图中未显示： **iscsigws** 是 iSCSI 网关。

6. 安装完成后，单击页面右下角的 *Complete* 按钮。这将打开一个窗口，显示命令 **ceph status** 的输出，以及控制面板访问信息。

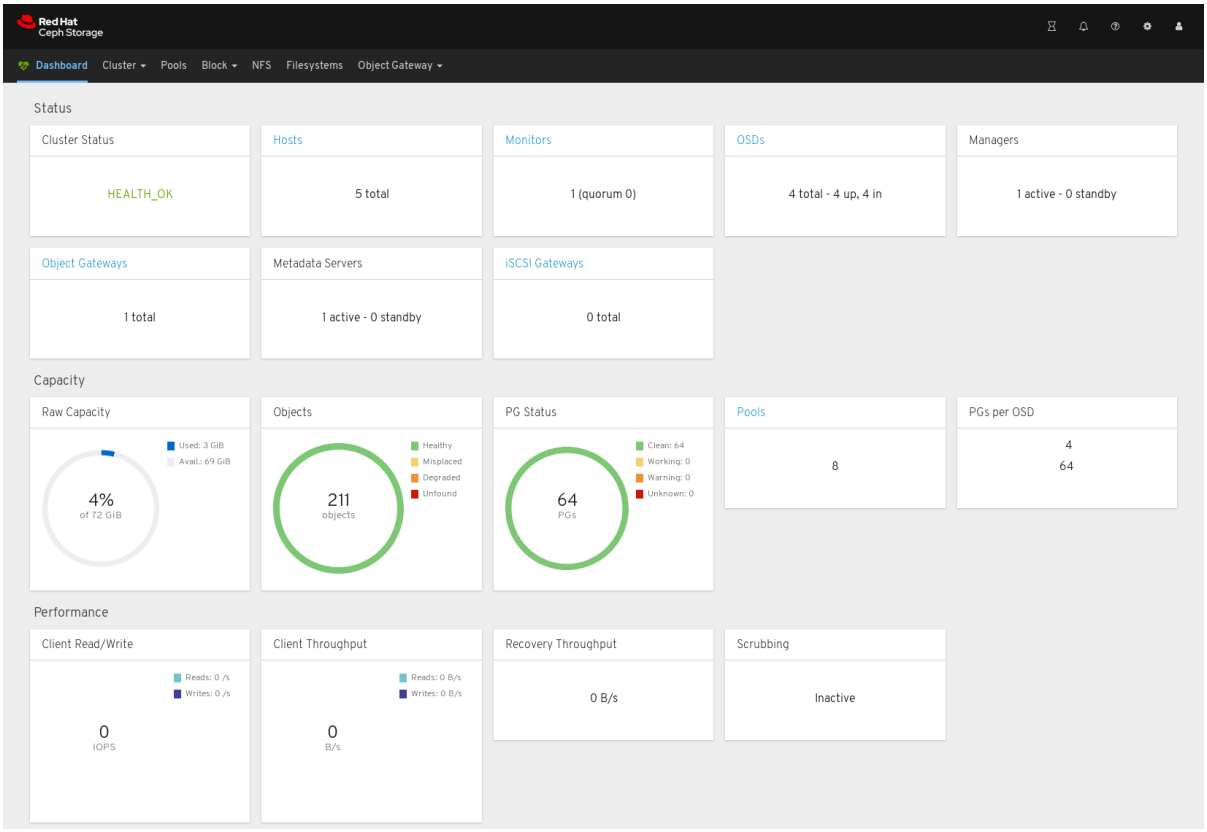


7. 将以下示例中的集群状态信息与集群中的集群状态信息进行比较。示例显示了一个健康集群，所有 OSD 都正常运行和内向，并且所有服务都处于活动状态。PG 处于 **active+clean** 状态。如果集群的某些方面不相同，请参阅故障排除指南以了解有关如何解决问题的信息。

8. 在 Ceph Cluster Status 窗口的底部，会显示控制面板访问信息，包括 URL、用户名和密码。记录此信息。



9. 使用上一步中的信息以及控制面板指南来访问控制面板。



控制面板提供了一个 Web 界面，您可以管理和监控 Red Hat Ceph Storage 集群。如需更多信息，请参阅控制面板指南。

10. 可选：查看 **cockpit-ceph-installer.log** 文件。此文件记录了选择的日志，以及探测进程生成的关联警告。它位于运行安装程序脚本 **ansible-runner-service.sh** 的用户主目录中。

第 5 章 使用 ANSIBLE 安装 RED HAT CEPH STORAGE

本章介绍如何使用 Ansible 应用来部署 Red Hat Ceph Storage 集群和其他组件，如元数据服务器或 Ceph 对象网关。

- 要安装 Red Hat Ceph Storage 集群，请参阅 第 5.2 节 “安装 Red Hat Ceph Storage 集群” 。
- 要安装元数据服务器，请参阅 第 5.4 节 “安装元数据服务器”。
- 要安装 **ceph-client** 角色，请参阅 第 5.5 节 “安装 Ceph 客户端角色”。
- 要安装 Ceph 对象网关，请参阅 第 5.6 节 “安装 Ceph 对象网关”。
- 若要配置多站点 Ceph 对象网关，请参阅 第 5.7 节 “配置多站点 Ceph 对象网关”。
- 要了解 Ansible 的 **--limit** 选项，请参阅 第 5.10 节 “了解 **limit** 选项”。

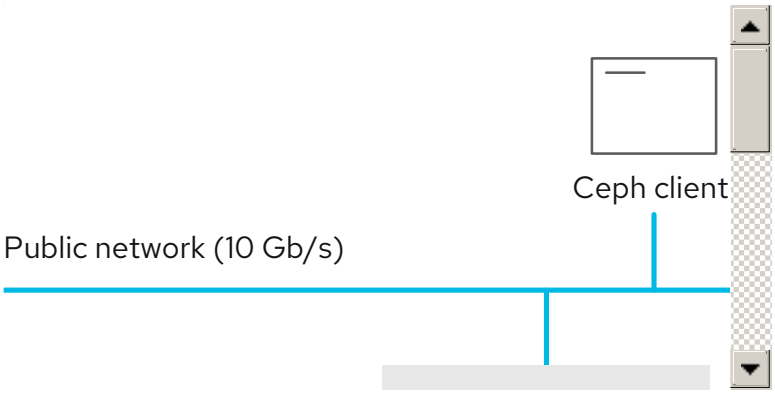
5.1. 先决条件

- 获取有效的客户订阅.
- 通过在每个节点上执行以下操作来准备集群节点：
  - 将节点注册到 Content Delivery Network (CDN) 并附加订阅 。
  - 启用适当的软件存储库。
  - 创建 Ansible 用户。
  - 启用免密码 SSH 访问。
  - （可选） 配置防火墙。

5.2. 安装 RED HAT CEPH STORAGE 集群

使用 Ansible 应用程序和 **ceph-ansible** playbook 在裸机或容器中安装 Red Hat Ceph Storage。在生产环境中使用 Ceph 存储集群时，必须至少有三个监控节点和三个 OSD 节点，其中包含多个 OSD 守护进程。生产环境中运行的典型 Ceph 存储群集通常包含十个或更多个节点。

在以下步骤中，从 Ansible 管理节点运行命令，除非另有指示。除非另有指定，否则此流程适用于裸机和容器部署。



重要

Ceph 可以使用一个监控器运行；但是，为了保证生产环境集群中的高可用性，红帽将仅支持具有至少三个 monitor 节点的部署。

重要

在 Red Hat Enterprise Linux 7.7 上的容器中部署 Red Hat Ceph Storage 4 将在 Red Hat Enterprise Linux 8 容器镜像上部署 Red Hat Ceph Storage 4。

先决条件

- 有效的客户订阅。
- 对 Ansible 管理节点的根级别访问权限.
- 用于 Ansible 应用的 **ansible** 用户帐户。
- 启用 Red Hat Ceph Storage Tools 和 Ansible 存储库
- 对于 **ISO** 安装，将最新的 ISO 镜像下载到 Ansible 节点上。请参阅 *Red Hat Ceph Storage 安装指南*中的*启用红帽 Ceph 存储存储库*一章中有关 *ISO 安装*的章节。

流程

1. 在 Ansible 管理节点上，作为 **root** 用户帐户登录。
2. 对于所有部署（**裸机** 或 **容器**），请安装 **ceph-ansible** 软件包：

   Red Hat Enterprise Linux 7

   ```
   [root@admin ~]# yum install ceph-ansible
   ```

   Red Hat Enterprise Linux 8

   ```
   [root@admin ~]# dnf install ceph-ansible
   ```

3. 进入 **/usr/share/ceph-ansible/** 目录：

   ```
   [root@admin ~]# cd /usr/share/ceph-ansible
   ```

4. 创建新的 **yml** 文件：

   ```
   [root@admin ceph-ansible]# cp group_vars/all.yml.sample group_vars/all.yml
   [root@admin ceph-ansible]# cp group_vars/osds.yml.sample group_vars/osds.yml
   ```

   a. **裸机**部署：

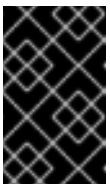
   ```
   [root@admin ceph-ansible]# cp site.yml.sample site.yml
   ```

   b. **容器**部署：

   ```
   [root@admin ceph-ansible]# cp site-container.yml.sample site-container.yml
   ```

5. 编辑新文件。

a. 打开以编辑 **group_vars/all.yml** 文件。



重要

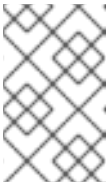
不支持使用自定义存储集群名称。不要将 **cluster** 参数设置为 **ceph** 以外的任何值。仅支持使用自定义存储集群名称的 Ceph 客户端，例如：**librados**、Ceph 对象网关和 RADOS 块设备镜像。

警告

默认情况下，Ansible 会尝试重启已安装但屏蔽的 **firewalld** 服务，这可能会导致 Red Hat Ceph Storage 部署失败。要临时解决这个问题，请在 **all.yml** 文件中将 **configure_firewall** 选项设置为 **false**。如果您正在运行 **firewalld** 服务，则不需要在 **all.yml** 文件中使用 **configure_firewall** 选项。

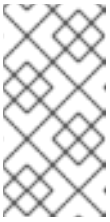
注意

将 **ceph_rhcs_version** 选项设置为 **4** 将引入最新版本的 Red Hat Ceph Storage 4。



注意

红帽建议在 **group_vars/all.yml** 文件中将 **dashboard_enabled** 选项设置为 **True**，而不要将它改为 **False**。如果要禁用仪表板，请参阅禁用 Ceph 仪表板。

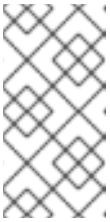


注意

与仪表板相关的组件已容器化。因此，对于 **Bare-metal** 或 **Container** 部署，必须包含 **ceph_docker_registry_username** 和 **ceph_docker_registry_password** 参数，以便 ceph-ansible 能够获取控制面板所需的容器镜像。

注意

如果您没有 Red Hat Registry Service Account，请使用 Registry Service Account 网页 创建一个。如需了解如何创建和管理令牌的详细信息，请参阅 *Red Hat Container Registry Authentication* 知识库。



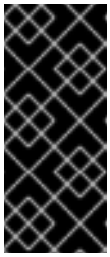
注意

除了将服务帐户用于 **ceph_docker_registry_username** 和 **ceph_docker_registry_password** 参数外，您还可以使用客户门户凭据，但若要确保安全性，可以对 **ceph_docker_registry_password** 参数进行加密。如需更多信息，请参阅使用 ansible-vault 加密 Ansible 密码变量。

i. CDN 安装的 **all.yml** 文件的 **裸机** 示例：

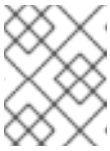
```
fetch_directory: ~/ceph-ansible-keys
ceph_origin: repository
ceph_repository: rhcs
ceph_repository_type: cdn
ceph_rhcs_version: 4
monitor_interface: eth0 (1)
public_network: 192.168.0.0/24
ceph_docker_registry: registry.redhat.io
ceph_docker_registry_auth: true
ceph_docker_registry_username: SERVICE_ACCOUNT_USER_NAME
ceph_docker_registry_password: TOKEN
dashboard_admin_user:
dashboard_admin_password:
node_exporter_container_image: registry.redhat.io/openshift4/ose-prometheus-node-exporter:v4.6
grafana_admin_user:
grafana_admin_password:
grafana_container_image: registry.redhat.io/rhceph/rhceph-4-dashboard-rhel8
prometheus_container_image: registry.redhat.io/openshift4/ose-prometheus:v4.6
alertmanager_container_image: registry.redhat.io/openshift4/ose-prometheus-alertmanager:v4.6
```

(1) 这是公共网络上的接口。



重要

从 Red Hat Ceph Storage 4.1 开始，您必须在 **/usr/share/ceph-ansible/group_vars/all.yml** 中取消注释或设置 **dashboard_admin_password** 和 **grafana_admin_password**。为每个用户设置安全密码.另外，为 **dashboard_admin_user** 和 **grafana_admin_user** 设置自定义用户名。



注意

对于 Red Hat Ceph Storage 4.2，如果您使用本地 registry 进行安装，请使用 4.6 作为 Prometheus 镜像标签。

ii. ISO 安装的 **all.yml** 文件的 **裸机** 示例：

```
fetch_directory: ~/ceph-ansible-keys
ceph_origin: repository
ceph_repository: rhcs
ceph_repository_type: iso
ceph_rhcs_iso_path: /home/rhceph-4-rhel-8-x86_64.iso
ceph_rhcs_version: 4
monitor_interface: eth0 (1)
public_network: 192.168.0.0/24
ceph_docker_registry: registry.redhat.io
ceph_docker_registry_auth: true
ceph_docker_registry_username: SERVICE_ACCOUNT_USER_NAME
ceph_docker_registry_password: TOKEN
```

```
dashboard_admin_user:
dashboard_admin_password:
node_exporter_container_image: registry.redhat.io/openshift4/ose-prometheus-
node-exporter:v4.6
grafana_admin_user:
grafana_admin_password:
grafana_container_image: registry.redhat.io/rhceph/rhceph-4-dashboard-rhel8
prometheus_container_image: registry.redhat.io/openshift4/ose-prometheus:v4.6
alertmanager_container_image: registry.redhat.io/openshift4/ose-prometheus-
alertmanager:v4.6
```

1 这是公共网络上的接口。

iii. **all.yml** 文件的**容器**示例：

```
fetch_directory: ~/ceph-ansible-keys
monitor_interface: eth0 (1)
public_network: 192.168.0.0/24
ceph_docker_image: rhceph/rhceph-4-rhel8
ceph_docker_image_tag: latest
containerized_deployment: true
ceph_docker_registry: registry.redhat.io
ceph_docker_registry_auth: true
ceph_docker_registry_username: SERVICE_ACCOUNT_USER_NAME
ceph_docker_registry_password: TOKEN
ceph_origin: repository
ceph_repository: rhcs
ceph_repository_type: cdn
ceph_rhcs_version: 4
dashboard_admin_user:
dashboard_admin_password:
node_exporter_container_image: registry.redhat.io/openshift4/ose-prometheus-
node-exporter:v4.6
grafana_admin_user:
grafana_admin_password:
grafana_container_image: registry.redhat.io/rhceph/rhceph-4-dashboard-rhel8
prometheus_container_image: registry.redhat.io/openshift4/ose-prometheus:v4.6
alertmanager_container_image: registry.redhat.io/openshift4/ose-prometheus-
alertmanager:v4.6
```

1 这是公共网络上的接口。

重要

查看*红帽生态系统目录*中的最新容器镜像标签，以安装最新的容器镜像，并应用所有最新的补丁。

iv. **all.yml** 文件的**容器**示例，当 Red Hat Ceph Storage 节点在部署过程中无法访问互联网：

```
fetch_directory: ~/ceph-ansible-keys
monitor_interface: eth0 (1)
public_network: 192.168.0.0/24
ceph_docker_image: rhceph/rhceph-4-rhel8
```

```
ceph_docker_image_tag: latest
containerized_deployment: true
ceph_docker_registry: LOCAL_NODE_FQDN:5000
ceph_docker_registry_auth: false
ceph_origin: repository
ceph_repository: rhcs
ceph_repository_type: cdn
ceph_rhcs_version: 4
dashboard_admin_user:
dashboard_admin_password:
node_exporter_container_image: LOCAL_NODE_FQDN:5000/openshift4/ose-prometheus-node-exporter:v4.6
grafana_admin_user:
grafana_admin_password:
grafana_container_image: LOCAL_NODE_FQDN:5000/rhceph/rhceph-4-dashboard-rhel8
prometheus_container_image: LOCAL_NODE_FQDN:5000/openshift4/ose-prometheus:4.6
alertmanager_container_image: LOCAL_NODE_FQDN:5000/openshift4/ose-prometheus-alertmanager:4.6
```

1 这是公共网络上的接口。

替换

- *LOCAL_NODE_FQDN*，使用您的本地主机 FQDN。

v. 从 Red Hat Ceph Storage 4.2 开始，**dashboard_protocol** 设置为 **https**，Ansible 将生成仪表板和 grafana 密钥和证书。对于自定义证书，在 **all.yml** 文件中，为 **裸机** 或 **容器** 部署更新 **dashboard_crt**、**dashboard_key**、**grafana_crt** 和 **grafana_key** 的 Ansible 安装程序主机的路径。

语法

```
dashboard_protocol: https
dashboard_port: 8443
dashboard_crt: 'DASHBOARD_CERTIFICATE_PATH'
dashboard_key: 'DASHBOARD_KEY_PATH'
dashboard_tls_external: false
dashboard_grafana_api_no_ssl_verify: "{{ True if dashboard_protocol == 'https' and not grafana_crt and not grafana_key else False }}"
grafana_crt: 'GRAFANA_CERTIFICATE_PATH'
grafana_key: 'GRAFANA_KEY_PATH'
```

b. 要使用可通过 http 或 https 代理访问的容器 registry 安装 Red Hat Ceph Storage，请在 **group_vars/all.yml** 文件中设置 **ceph_docker_http_proxy** 或 **ceph_docker_https_proxy** 变量。

示例

```
ceph_docker_http_proxy: http://192.168.42.100:8080
ceph_docker_https_proxy: https://192.168.42.100:8080
```

如果您需要排除代理配置的一些主机，请使用 **group_vars/all.yml** 文件中的 **ceph_docker_no_proxy** 变量。

示例

```
ceph_docker_no_proxy: "localhost,127.0.0.1"
```

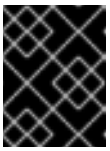
c. 除了为 Red Hat Ceph Storage 的代理安装编辑 **all.yml** 文件外，编辑 **/etc/environment** 文件：

示例

```
HTTP_PROXY: http://192.168.42.100:8080
HTTPS_PROXY: https://192.168.42.100:8080
NO_PROXY: "localhost,127.0.0.1"
```

这会触发 podman 启动容器化服务，如 prometheus、grafana-server、alertmanager 和 node-exporter，并下载所需的镜像。

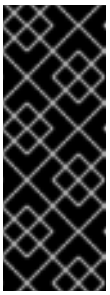
d. 对于 **裸机** 或 **容器**中的所有部署，请编辑 **group_vars/osds.yml** 文件。



重要

不要在安装操作系统的设备上安装 OSD。在操作系统和 OSD 之间共享相同的设备会导致性能问题。

Ceph-ansible 使用 **ceph-volume** 工具准备存储设备，供 Ceph 使用。您可以将 **osds.yml** 配置为以不同的方式使用存储设备，以优化特定工作负载的性能。



重要

以下示例使用 BlueStore 对象存储，即 Ceph 用于存储设备上数据的格式。在以前的版本中，Ceph 使用 FileStore 作为对象存储。这种格式对于新的 Red Hat Ceph Storage 4.0 安装已弃用，因为 BlueStore 提供了更多功能和更高的性能。虽然仍可使用 FileStore，但使用它需要红帽支持例外。如需有关 BlueStore 的更多信息，请参阅 *Red Hat Ceph Storage 架构指南* 中的 Ceph BlueStore。

i. **自动发现**

```
osd_auto_discovery: true
```

上例使用系统上的所有空存储设备来创建 OSD，因此您不必显式指定它们。**ceph-volume** 工具检查是否有空设备，因此不会使用不为空的设备。

注意

如果稍后决定使用 **purge-docker-cluster.yml** 或 **purge-cluster.yml** 来删除集群，您必须注释掉 **osd_auto_discovery**，并声明 **osds.yml** 文件中的 OSD 设备。如需更多信息，请参阅 Ansible 部署的存储集群。

ii. **简单配置**

第一个场景

```
devices:
  - /dev/sda
  - /dev/sdb
```

或

第二个场景

```
devices:
  - /dev/sda
  - /dev/sdb
  - /dev/nvme0n1
  - /dev/sdc
  - /dev/sdd
  - /dev/nvme1n1
```

或

第三个场景

```
lvm_volumes:
   - data: /dev/sdb
   - data: /dev/sdc
```

或

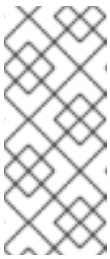
第四个场景

```
lvm_volumes:
    - data: /dev/sdb
    - data:/dev/nvme0n1
```

当仅使用 **devices** 选项时，**ceph-volume lvm batch** 模式会自动优化 OSD 配置。

在第一个场景中，如果 **devices** 是传统的硬盘驱动器或 SSD，则每个设备会创建一个 OSD。

在第二种场景中，如果结合了传统的硬盘驱动器和 SSD，数据将放置在传统的硬盘驱动器（**sda**、**sdb**）上，并且将 BlueStore 数据库尽可能在 SSD（**nvme0n1**）上创建。同样，无论上述设备顺序如何，数据都放置在传统的硬盘驱动器（**sdc**、**sdd**）上，而 BlueStore 数据库则在 SSD **nvme1n1** 上创建。



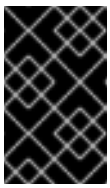
注意

默认情况下，**ceph-ansible** 不会覆盖 **bluestore_block_db_size** 和 **bluestore_block_wal_size** 的默认值。您可以使用 **group_vars/all.yml** 文件中的 **ceph_conf_overrides** 设置 **bluestore_block_db_size**。**bluestore_block_db_size** 的值应大于 2 GB。

在第三个场景中，数据放置在传统硬盘驱动器（**sdb**、**sdc**）上，而 BlueStore 数据库则

在第三个场景中，数据放置在传统硬盘驱动器（**sdb**、**sdc**）上，而 BlueStore 数据库则在同一设备上并置。

在第四个场景中，数据放置在传统的硬盘驱动器 (**sdb**) 和 SSD (**nvme1n1**) 上，并且 BlueStore 数据库同时驻留在同一设备上。这不同于使用 **devices** 指令，将 BlueStore 数据库放置在 SSD 上。



重要

ceph-volume lvm batch mode 通过将数据放置在传统硬盘驱动器上以及 SSD 上的 BlueStore 数据库来创建优化的 OSD 配置。如果要指定要使用的逻辑卷和卷组，您可以直接按照以下*高级配置*场景创建它们。

iii. **高级配置**

第一个场景

```
devices:
  - /dev/sda
  - /dev/sdb
dedicated_devices:
  - /dev/sdx
  - /dev/sdy
```

或

第二个场景

```
devices:
  - /dev/sda
  - /dev/sdb
dedicated_devices:
  - /dev/sdx
  - /dev/sdy
bluestore_wal_devices:
  - /dev/nvme0n1
  - /dev/nvme0n2
```

在第一种场景中，有两个 OSD。**sda** 和 **sdb** 设备各自都有自己的数据片段和 WAL（write-ahead logs）。额外的字典 **dedicated_devices** 用于分别在 **sdx** 和 **sdy** 上隔离其数据库，也称为 **block.db**。

在第二个场景中，使用另一个额外的字典 **bluestore_wal_devices** 来隔离 NVMe devices **nvme0n1** 和 **nvme0n2** 上的 WAL。通过使用 **devices**、**dedicated_devices** 和 **bluestore_wal_devices** 选项，您可以将 OSD 的所有组件隔离到单独的设备上。像这样布置 OSD 可以提高整体性能。

iv. **预先创建的逻辑卷**

第一个场景

```
lvm_volumes:
  - data: data-lv1
    data_vg: data-vg1
    db: db-lv1
```

```
    db_vg: db-vg1
    wal: wal-lv1
    wal_vg: wal-vg1
  - data: data-lv2
    data_vg: data-vg2
    db: db-lv2
    db_vg: db-vg2
    wal: wal-lv2
    wal_vg: wal-vg2
```

或

第二个场景

```
lvm_volumes:
  - data: /dev/sdb
    db:    db-lv1
    db_vg: db-vg1
    wal: wal-lv1
    wal_vg: wal-vg1
```

默认情况下，Ceph 使用逻辑卷管理器在 OSD 设备上创建逻辑卷。在上述*简单配置*和*高级配置*示例中，Ceph 会自动在设备上创建逻辑卷。您可以通过指定 **lvm_volumes** 字典，将之前创建的逻辑卷用于 Ceph。

在第一种场景中，数据放置在专用逻辑卷、数据库和 WAL 上。您还可以仅指定数据、数据和 WAL，或数据和数据库。**data:** 行必须指定要存储数据的逻辑卷名称，**data_vg:** 必须指定包含数据逻辑卷的卷组的名称。同样，**db:** 用于指定数据库存储在 上的逻辑卷，而 **db_vg:** 用于指定其逻辑卷所在的卷组。**wal:** 行指定 WAL 存储的逻辑卷，而 **wal_vg:** 行则指定包含它的卷组。

在第二种场景中，为 **data:** 选项设置实际设备名称，这样做不需要指定 **data_vg:** 选项。您必须为 BlueStore 数据库和 WAL 设备指定逻辑卷名称和卷组详情。

重要

使用 **lvm_volumes：** 必须事先创建卷组和逻辑卷。**ceph-ansible** 不会创建卷组和逻辑卷。

注意

如果使用所有 NVMe SSD，则设置 **osds_per_device: 2**。如需更多信息，请参阅 *Red Hat Ceph Storage 安装指南* 中*为所有 NVMe 存储配置 OSD Ansible 设置*。

注意

重启 Ceph OSD 节点后，块设备分配可能会改变。例如，**sdc** 可能成为 **sdd**。您可以使用持久的命名设备，如 **/dev/disk/by-path/** 设备路径，而不是传统的块设备名称。

6. 对于**裸机**或**容器**中的所有部署，请创建 Ansible 清单文件，然后打开它进行编辑：

```
[root@admin ~]# cd /usr/share/ceph-ansible/
[root@admin ceph-ansible]# touch hosts
```

相应地编辑 **hosts** 文件。

注意

有关编辑 Ansible 清单位置的信息，*请参阅配置 Ansible 清单位置*。

a. 在 **[grafana-server]** 下添加一个节点。此角色安装 Grafana 和 Prometheus，以提供 Ceph 集群性能的实时洞察。这些指标显示在 Ceph 控制面板中，您可以通过它来监控和管理集群。需要安装仪表板、Grafana 和 Prometheus。您可以在 Ansible 管理节点上并置指标功能。如果这样做，请确保节点的系统资源大于独立指标节点所需的值。

```
[grafana-server]
GRAFANA-SERVER_NODE_NAME
```

b. 在 **[mons]** 部分添加 monitor 节点：

```
[mons]
MONITOR_NODE_NAME_1
MONITOR_NODE_NAME_2
MONITOR_NODE_NAME_3
```

c. 在 **[osds]** 部分下添加 OSD 节点：

```
[osds]
OSD_NODE_NAME_1
OSD_NODE_NAME_2
OSD_NODE_NAME_3
```

注意

如果节点名称是按照数字有顺序的，您可以在节点名称的末尾添加一个范围指定符 (**[1:10]**)。例如：

```
[osds]
example-node[1:10]
```

注意

对于新安装的 OSD，默认的对象存储格式为 BlueStore。

d. 另外，在**容器**部署中，通过在 **[mon]** 和 **[osd]** 部分下添加相同的节点，将 Ceph 监控守护进程与一个节点上的 Ceph OSD 守护进程共存。有关更多信息，请参见下面*附加资源*部分中的 Ceph 守护进程链接。

e. 在 **[mgrs]** 部分下，添加 Ceph Manager (**ceph-mgr**) 节点。这会将 Ceph 管理器守护进程与 Ceph 监控守护进程共存。

```
[mgrs]
MONITOR_NODE_NAME_1
```

```
MONITOR_NODE_NAME_2
MONITOR_NODE_NAME_3
```

7. 另外，如果您想要使用主机特定参数，对于所有部署（**裸机**或**容器**中），创建 **host_vars** 目录，其中包含主机文件，使其包含特定于主机的任何参数。

   a. 创建 **host_vars** 目录：

   ```
   [ansible@admin ~]$ mkdir /usr/share/ceph-ansible/host_vars
   ```

   b. 进入 **host_vars** 目录：

   ```
   [ansible@admin ~]$ cd /usr/share/ceph-ansible/host_vars
   ```

   c. 创建主机文件。将 *host-name-short-name* 格式用于文件名称，例如：

   ```
   [ansible@admin host_vars]$ touch tower-osd6
   ```

   d. 使用任何主机特定参数更新该文件，例如：

      i. 在**裸机**部署中，使用 **devices** 参数指定 OSD 节点要使用的设备。当 OSD 使用具有不同名称的设备或其中一个设备在其中一个 OSD 上出现故障时，使用 **devices** 会比较有用。

      ```
      devices:
        DEVICE_1
        DEVICE_2
      ```

      示例

      ```
      devices:
        /dev/sdb
        /dev/sdc
      ```

      注意

      在指定没有设备时，将 **group_vars/osds.yml** 文件中的 **osd_auto_discovery** 参数设置为 **true**。

8. 另外，对于所有部署、**裸机或 容器** 中，您可以使用 Ceph Ansible 创建自定义 CRUSH 层次结构：

   a. 设置 Ansible 清单文件。使用 **osd_crush_location** 参数，指定 OSD 主机处于 CRUSH map 的层次结构中的位置。您必须指定至少两种 CRUSH bucket 类型来指定 OSD 的位置，一种 bucket 类型 必须是 host。默认情况下，包括 **root**, **datacenter**, **room**, **row**, **pod**, **pdu**, **rack**, **chassis** 和 **host**。

   语法

   ```
   [osds]
   CEPH_OSD_NAME osd_crush_location="{ 'root': ROOT_BUCKET_', 'rack':
   'RACK_BUCKET', 'pod': 'POD_BUCKET', 'host': 'CEPH_HOST_NAME' }"
   ```

   示例

```
[osds]
ceph-osd-01 osd_crush_location="{ 'root': 'default', 'rack': 'rack1', 'pod': 'monpod', 'host':
'ceph-osd-01' }"
```

b. 编辑 **group_vars/osds.yml** 文件，并将 **crush_rule_config** 和 **create_crush_tree** 参数设置为 **True**。如果您不想使用默认的 CRUSH 规则，请至少创建一个 CRUSH 规则，例如：

```
crush_rule_config: True
crush_rule_hdd:
    name: replicated_hdd_rule
    root: root-hdd
    type: host
    class: hdd
    default: True
crush_rules:
  - "{{ crush_rule_hdd }}"
create_crush_tree: True
```

如果您使用更快的 SSD 设备，请按如下所示编辑参数：

```
crush_rule_config: True
crush_rule_ssd:
    name: replicated_ssd_rule
    root: root-ssd
    type: host
    class: ssd
    default: True
crush_rules:
  - "{{ crush_rule_ssd }}"
create_crush_tree: True
```

注意

如果未部署 **ssd** 和 **hdd** OSD，默认 CRUSH 规则会失败，因为默认规则现在包含 类**参数**，必须定义。

c. 创建池，并在 **group_vars/clients.yml** 文件中创建 **crush_rules** ：

示例

```
copy_admin_key: True
user_config: True
pool1:
  name: "pool1"
  pg_num: 128
  pgp_num: 128
  rule_name: "HDD"
  type: "replicated"
```

```
  device_class: "hdd"
pools:
  - "{{ pool1 }}"
```

d. 查看树：

```
[root@mon ~]# ceph osd tree
```

e. 验证池：

```
[root@mon ~]# for i in $(rados lspools); do echo "pool: $i"; ceph osd pool get $i
crush_rule; done

pool: pool1
crush_rule: HDD
```

9. 对于所有部署，裸机或容器中，请使用 **ansible** 用户登录或切换到该容器。

a. 创建 **ceph-ansible-keys** 目录，其中 **Ansible** 存储 **ceph-ansible playbook** 生成的临时值：

```
[ansible@admin ~]$ mkdir ~/ceph-ansible-keys
```

b. 进入 **/usr/share/ceph-ansible/** 目录：

```
[ansible@admin ~]$ cd /usr/share/ceph-ansible/
```

c. 验证 **Ansible** 能否访问 **Ceph** 节点：

```
[ansible@admin ceph-ansible]$ ansible all -m ping -i hosts
```

10. 运行 **ceph-ansible playbook**。

a. 裸机部署：

```
[ansible@admin ceph-ansible]$ ansible-playbook site.yml -i hosts
```

b.

容器部署：

```
[ansible@admin ceph-ansible]$ ansible-playbook site-container.yml -i hosts
```

注意

如果您将 Red Hat Ceph Storage 部署到 Red Hat Enterprise Linux Atomic Host 主机，请使用 --skip-tags=with_pkg 选项：

```
[user@admin ceph-ansible]$ ansible-playbook site-container.yml --skip-tags=with_pkg -i hosts
```

注意

要提高部署速度，请在 ansible-playbook 中使用 --forks 选项。默认情况下，ceph-ansible 将 fork 设置为 20。在这个版本中，最多 20 个节点将同时安装。要一次安装最多 30 个节点，请运行 ansible-playbook --forks 30 *PLAYBOOK FILE* -i hosts。必须监控管理节点上的资源，以确保它们不会被过度使用。如果是，则减少传递给 --forks 的数字。

11.

等待 Ceph 部署完成。

输出示例

```
INSTALLER STATUS ******************************
Install Ceph Monitor           : Complete (0:00:30)
Install Ceph Manager           : Complete (0:00:47)
Install Ceph OSD               : Complete (0:00:58)
Install Ceph RGW               : Complete (0:00:34)
Install Ceph Dashboard         : Complete (0:00:58)
Install Ceph Grafana           : Complete (0:00:50)
Install Ceph Node Exporter     : Complete (0:01:14)
```

12.

验证 Ceph 存储集群的状态。

a.

裸机部署：

```
[root@mon ~]# ceph health
HEALTH_OK
```

b. **容器部署：**

Red Hat Enterprise Linux 7

```
[root@mon ~]# docker exec ceph-mon-ID ceph health
```

Red Hat Enterprise Linux 8

```
[root@mon ~]# podman exec ceph-mon-ID ceph health
```

替换

- **使用 Ceph 监控节点的主机名替换 *ID*：**

示例

```
[root@mon ~]# podman exec ceph-mon-mon0 ceph health
HEALTH_OK
```

13. **对于裸机或容器中的所有部署，使用 rados 验证存储集群是否正常工作。**

a. **从 Ceph 监控节点，创建具有八个放置组 (PG) 的测试池：**

语法

```
[root@mon ~]# ceph osd pool create POOL_NAME PG_NUMBER
```

示例

```
[root@mon ~]# ceph osd pool create test 8
```

b. **创建名为 hello-world.txt 的文件：**

语法

```
[root@mon ~]# vim FILE_NAME
```

示例

```
[root@mon ~]# vim hello-world.txt
```

c. **使用对象名称 hello-world 将 hello-world.txt 上传到测试池中：**

语法

```
[root@mon ~]# rados --pool POOL_NAME put OBJECT_NAME OBJECT_FILE_NAME
```

示例

```
[root@mon ~]# rados --pool test put hello-world hello-world.txt
```

d. **从 test 池下载 hello-world，保存为 fetch.txt ：**

语法

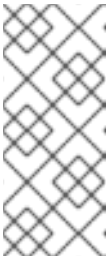
```
[root@mon ~]# rados --pool POOL_NAME get OBJECT_NAME OBJECT_FILE_NAME
```

示例

```
[root@mon ~]# rados --pool test get hello-world fetch.txt
```

e. **检查 fetch.txt 的内容：**

```
[root@mon ~]# cat fetch.txt
"Hello World!"
```



注意

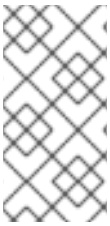
除了验证存储集群状态外，您还可以使用 ceph-medic 工具来全面诊断 Ceph 存储集群。请参阅 Red Hat Ceph Storage 4 *故障排除指南*中的*安装和使用 ceph-medic 来诊断 Ceph 存储集群*章节。

其它资源

- 常见 Ansible 设置列表。
- 常见 OSD 设置列表。
- 详情请参阅*共存容器化 Ceph 守护进程*。

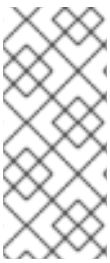
5.3. 为所有 NVME 存储配置 OSD ANSIBLE 设置

要提高整体性能，您可以将 Ansible 配置为仅使用非易失性内存表达 (NVMe) 设备进行存储。通常，每个设备仅配置一个 OSD，这可以充分利用 NVMe 设备潜在的吞吐量。



注意

如果混合使用了 SSD 和 HDD，则 SSD 将用于数据库，或者 block.db，而不是用于 OSD 中的数据。



注意

在测试过程中，发现每个 NVMe 设备上配置两个 OSD 可提供最佳性能。红帽建议将 osds_per_device 选项设置为 2，但这不是强制要求。其他值可能会在您的环境中提供更好的性能。

先决条件

- 访问 Ansible 管理节点.
- 安装 ceph-ansible 软件包.

流程

流程

1. **在 group_vars/osds.yml 中设置 osds_per_device: 2:**

```
osds_per_device: 2
```

2. **列出 devices 中的 NVMe 设备：**

```
devices:
  - /dev/nvme0n1
  - /dev/nvme1n1
  - /dev/nvme2n1
  - /dev/nvme3n1
```

3. **group_vars/osds.yml 中的设置类似以下示例：**

```
osds_per_device: 2
devices:
  - /dev/nvme0n1
  - /dev/nvme1n1
  - /dev/nvme2n1
  - /dev/nvme3n1
```

注意

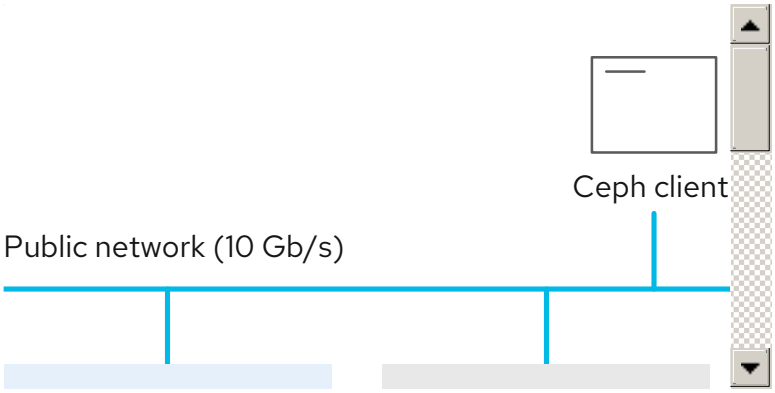
您必须将 devices 用于此配置，而不是使用 lvm_volumes。这是因为 lvm_volumes 通常与预先创建的逻辑卷一起使用，而 osds_per_device 则表示 Ceph 自动创建逻辑卷。

其它资源

- **如需了解更多详细信息，请参阅 *Red Hat Ceph Storage 安装指南*中的安装 *Red Hat Ceph Storage 集群*。**

5.4. 安装元数据服务器

使用 Ansible 自动化应用安装 Ceph 元数据服务器 (MDS)。元数据服务器守护进程是部署 Ceph 文件系统所必需的。



先决条件

- **一个正常工作的 Red Hat Ceph Storage 集群。**
- **启用免密码 SSH 访问。**

流程

在 Ansible 管理节点上执行下列步骤。

1. **在 /etc/ansible/hosts 文件中添加新部分 [mdss] ：**

```
[mdss]
MDS_NODE_NAME1
MDS_NODE_NAME2
MDS_NODE_NAME3
```

将 *MDS_NODE_NAME* 替换为您要安装 Ceph 元数据服务器的节点的主机名。

或者，您可以通过在 [osds] 和 [mdss] 部分下添加同一节点，将元数据服务器与 OSD 守护进程并置。

2. **进入 /usr/share/ceph-ansible 目录：**

```
[root@admin ~]# cd /usr/share/ceph-ansible
```

3. **（可选）您可以更改默认变量。**

a. 创建名为 **mdss.yml** 的 **group_vars/mdss.yml** 的副本：

```
[root@admin ceph-ansible]# cp group_vars/mdss.yml.sample group_vars/mdss.yml
```

b. （可选）编辑 **inmdss.yml** 中的参数。详情请查看 **mdss.yml**。

4. 以 **ansible** 用户身份，运行 **Ansible playbook:**

- **裸机部署：**

```
[user@admin ceph-ansible]$ ansible-playbook site.yml --limit mdss -i hosts
```

- **容器部署：**

```
[ansible@admin ceph-ansible]$ ansible-playbook site-container.yml --limit mdss -i hosts
```

5. 安装元数据服务器后，您现在可以配置它们。详情请参阅《红帽 *Ceph 存储文件系统指南》中的 Ceph 文件系统元数据服务器* 章节。

其它资源

- Red Hat Ceph Storage 4 的 *Ceph 文件系统指南*
- 详情请参阅*共存容器化 Ceph 守护进程*。
- 详情请参阅 *了解 限制 选项*。

5.5. 安装 CEPH 客户端角色

ceph-ansible 实用程序提供 **ceph-client** 角色，将 Ceph 配置文件和管理密钥环复制到节点。此外，您还可以使用此角色创建自定义池和客户端。

先决条件

- 正在运行的 Ceph 存储集群，最好处于 active + clean 状态。
- 执行要求中列出的任务。
- 启用免密码 SSH 访问。

流程

在 Ansible 管理节点上执行下列任务：

1. 在 /etc/ansible/hosts 文件中添加新部分 [clients]：

```
[clients]
CLIENT_NODE_NAME
```

将 *CLIENT_NODE_NAME* 替换为您要安装 ceph-client 角色的节点的主机名。

2. 进入 /usr/share/ceph-ansible 目录：

```
[root@admin ~]# cd /usr/share/ceph-ansible
```

3. 为 clients.yml.sample 文件创建一个新副本，取名为 client.yml：

```
[root@admin ceph-ansible ~]# cp group_vars/clients.yml.sample group_vars/clients.yml
```

4. 打开 group_vars/clients.yml 文件，取消注释以下行：

```
keys:
  - { name: client.test, caps: { mon: "allow r", osd: "allow class-read object_prefix
rbd_children, allow rwx pool=test" },  mode: "{{ ceph_keyring_permissions }}" }
```

a. 使用实际客户端名称替换 client.test，并在客户端定义行中添加客户端密钥，例如：

```
key: "ADD-KEYRING-HERE=="
```

现在，整行示例类似如下：

```
- { name: client.test, key: "AQAin8tUMICVFBAALRHNrV0Z4MXupRw4v9JQ6Q==", caps:
{ mon: "allow r", osd: "allow class-read object_prefix rbd_children, allow rwx pool=test" },
mode: "{{ ceph_keyring_permissions }}" }
```

注意

ceph-authtool --gen-print-key 命令可以生成新的客户端密钥。

5. （可选）指示 **ceph-client** 创建池和客户端。

   a. 更新 **clients.yml**。

   - 取消注释 **user_config** 设置并将其设置为 **true**。
   - 取消注释 **pools** 和 **keys** 部分，并根据需要进行更新。您可以使用 **cephx** 功能定义自定义池和客户端名称。

   b. 将 **osd_pool_default_pg_num** 设置添加到 **all.yml** 文件的 **ceph_conf_overrides** 部分：

```
ceph_conf_overrides:
  global:
    osd_pool_default_pg_num: NUMBER
```

   将 ***NUMBER*** 替换为 **PG** 的默认数量。

6. 以 **ansible** 用户身份，运行 **Ansible playbook:**

   a. 裸机部署：

```
[ansible@admin ceph-ansible]$ ansible-playbook site.yml --limit clients -i hosts
```

b. **容器部署：**

```
[ansible@admin ceph-ansible]$ ansible-playbook site-container.yml --limit clients -i hosts
```

其它资源

- 详情请参阅 *了解 限制 选项*。

5.6. 安装 CEPH 对象网关

Ceph 对象网关（也称为 RADOS 网关）是在 librados API 基础上构建的对象存储接口，为应用提供 Ceph 存储集群的 RESTful 网关。

先决条件

- 正在运行一个 Red Hat Ceph Storage 集群，最好处于 active + clean 状态。
- 启用免密码 SSH 访问。
- 在 Ceph 对象网关节点上，执行 第 3 章 *安装 Red Hat Ceph Storage 的要求* 中列出的任务。

警告

如果您要在多站点配置中使用 Ceph 对象网关，则仅完成第 1 - 6 步。在配置多站点前不要运行 Ansible playbook，因为这将在单个站点配置中启动对象网关。Ansible 在单个站点配置中启动后，无法将网关重新配置为多站点设置。完成第 1 到 6 步后，继续配置多站点 Ceph 对象网关部分以设置多站点。

流程

在 Ansible 管理节点上执行下列任务：

1.

将网关主机添加到 [rgws] 部分下的 /etc/ansible/hosts 文件中，以将其角色标识到 Ansible。如果主机有顺序命名，请使用范围，例如：

```
[rgws]
<rgw_host_name_1>
<rgw_host_name_2>
<rgw_host_name[3..10]>
```

2. 进入 Ansible 配置目录：

```
[root@ansible ~]# cd /usr/share/ceph-ansible
```

3. 从示例文件创建 rgws.yml 文件：

```
[root@ansible ~]# cp group_vars/rgws.yml.sample group_vars/rgws.yml
```

4. 打开并编辑 group_vars/rgws.yml 文件。要将管理员密钥复制到 Ceph 对象网关节点，取消注释 copy_admin_key 选项：

```
copy_admin_key: true
```

5. 在 all.yml 文件中，必须指定一个 radosgw_interface。

```
radosgw_interface: <interface>
```

替换：

- 使用 Ceph 对象网关节点侦听的接口替换 <interface>

例如：

```
radosgw_interface: eth0
```

指定该接口可防止 Civetweb 在同一主机上运行多个实例时绑定到与另一个 Civetweb 实例相同的 IP 地址。

如需了解更多详细信息，请参阅 **all.yml** 文件。

6.

通常，要更改默认设置，取消注释 **rgws.yml** 文件中的设置，并相应地进行更改。要对不在 **rgws.yml** 文件中的设置进行其他更改，请在 **all.yml** 文件中使用 **ceph_conf_overrides:**。

```
ceph_conf_overrides:
  client.rgw.rgw1:
    rgw_override_bucket_index_max_shards: 16
    rgw_bucket_default_quota_max_objects: 1638400
```

如需高级配置详细信息，请参阅 **Red Hat Ceph Storage 4** ***Ceph Object Gateway for Production*** **指南。高级议题包括：**

- ***配置 Ansible 组***
- ***开发存储策略*****。如需有关如何创建和配置池的更多详细信息，请参阅*** *创建根池*、*创建系统池***和** *创建数据放置策略***部分。**

  如需存储桶分片的配置详情，请参阅 Bucket Sharding。

7.

运行 **Ansible playbook:**

警告

如果要设置多站点，请不要运行 Ansible playbook。继续配置多站点 Ceph 对象网关部分来设置多站点。

a.

裸机部署：

```
[user@admin ceph-ansible]$ ansible-playbook site.yml --limit rgws -i hosts
```

.

b.

容器部署：

```
[user@admin ceph-ansible]$ ansible-playbook site-container.yml --limit rgws -i hosts
```

注意

Ansible 确保每个 Ceph 对象网关正在运行。

对于单一站点配置，请将 Ceph 对象网关添加到 Ansible 配置。

对于多站点部署，您应该为每个区域都有一个 Ansible 配置。也就是说，Ansible 将为该区创建一个 Ceph 存储集群和网关实例。

安装完多站点群集后，请继续 Red Hat Ceph Storage 4 *对象网关指南*中的多站点章节，以获取为多站点配置集群的详细信息。

其它资源

- **详情请参阅 *了解 限制 选项*。**
- **Red Hat Ceph Storage 4 *对象网关指南***

5.7. 配置多站点 CEPH 对象网关

作为系统管理员，您可以配置多站点 Ceph 对象网关来镜像集群间的数据，以满足灾难恢复的需要。

您可以使用一个或多个 RGW 域来配置多站点。域允许其中的 RGW 独立于域外的 RGW 并与其隔离。这样，一个域中写入 RGW 的数据就无法被另一个域中的 RGW 访问。

警告

在单一站点配置中已使用网关后，Ceph-ansible 无法将网关重新配置为多站点设置。您可以手动部署此配置。联系*红帽支持*以获取帮助。

注意

从 Red Hat Ceph Storage 4.1，您不需要在 group_vars/all.yml 文件中设置 rgw_multisite_endpoints_list 的值。

如需更多信息，请参阅 *Red Hat Ceph Storage Object Gateway Configuration and Administration Guide* 中的 *多站点* 部分。

5.7.1. 先决条件

- 两个 Red Hat Ceph Storage 集群。
- 在 Ceph 对象网关节点上，执行 *Red Hat Ceph Storage安装指南*中的*安装 Red Hat Ceph Storage 要求*一节中列出的任务。
- 对于每个对象网关节点，执行 *Red Hat Ceph Storage 安装指南*中的*安装 Ceph 对象网关*一节中的第 1 到 6 步。

5.7.2. 使用一个域配置多站点 Ceph 对象网关

Ceph-ansible 配置 Ceph 对象网关，以在具有多个 Ceph 对象网关实例的多个存储集群之间镜像数据。

警告

在单一站点配置中已使用网关后，Ceph-ansible 无法将网关重新配置为多站点设置。您可以手动部署此配置。联系*红帽支持*以获取帮助。

先决条件

- 两个正在运行的 **Red Hat Ceph Storage 集群。**
- 在 **Ceph 对象网关节点上，执行 *Red Hat Ceph Storage安装指南*中的*安装 Red Hat Ceph Storage 要求*一节中列出的任务。**
- 对于每个对象网关节点，执行 ***Red Hat Ceph Storage 安装指南*中的*安装 Ceph 对象网关*一节中的第 1 到 6 步。**

流程

1. **生成系统密钥并将其输出捕获至 multi-site-keys.txt 文件中：**

```
[root@ansible ~]# echo system_access_key: $(cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 20 | head -n 1) > multi-site-keys.txt
[root@ansible ~]# echo system_secret_key: $(cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 40 | head -n 1) >> multi-site-keys.txt
```

主存储集群

a. **进入 Ceph-ansible 配置目录：**

```
[root@ansible ~]# cd /usr/share/ceph-ansible
```

b. **打开并编辑 group_vars/all.yml 文件。取消注释 rgw_multisite 行并将其设置为 true。取消注释 rgw_multisite_proto 参数。**

```
rgw_multisite: true
rgw_multisite_proto: "http"
```

c.

在 /usr/share/ceph-ansible 中创建 host_vars 目录：

```
[root@ansible ceph-ansible]# mkdir host_vars
```

d.

在 host_vars 中为主存储群集上的每个对象网关节点创建一个文件。文件名应当与 Ansible 清单文件中使用的名称相同。例如，如果对象网关节点命名为 rgw-primary，则创建 host_vars/rgw-primary 文件。

语法

```
touch host_vars/NODE_NAME
```

示例

```
[root@ansible ceph-ansible]# touch host_vars/rgw-primary
```

注意

如果集群中有多个 Ceph 对象网关节点用于多站点配置，则为每个节点创建单独的文件。

e.

编辑该文件，并添加对应对象网关节点上所有实例的配置详情。配置以下设置，并相应地更新 *ZONE_NAME*、*ZONE_GROUP_NAME*、*ZONE_USER_NAME*、*ZONE_DISPLAY_NAME* 和 *REALM_NAME*。使用 multi-site-keys.txt 文件中保存的随机字符串用于 *ACCESS_KEY* 和 *SECRET_KEY*。

语法

```
rgw_instances:
  - instance_name: 'INSTANCE_NAME'
```

```
    rgw_multisite: true
    rgw_zonemaster: true
    rgw_zonesecondary: false
    rgw_zonegroupmaster: true
    rgw_zone: ZONE_NAME_1
    rgw_zonegroup: ZONE_GROUP_NAME_1
    rgw_realm: REALM_NAME_1
    rgw_zone_user: ZONE_USER_NAME_1
    rgw_zone_user_display_name: "ZONE_DISPLAY_NAME_1"
    system_access_key: ACCESS_KEY_1
    system_secret_key: SECRET_KEY_1
    radosgw_address: "{{ _radosgw_address }}"
    radosgw_frontend_port: RGW_PRIMARY_PORT_NUMBER_1
```

示例

```
rgw_instances:
  - instance_name: 'rgw0'
    rgw_multisite: true
    rgw_zonemaster: true
    rgw_zonesecondary: false
    rgw_zonegroupmaster: true
    rgw_zone: paris
    rgw_zonegroup: idf
    rgw_realm: france
    rgw_zone_user: jacques.chirac
    rgw_zone_user_display_name: "Jacques Chirac"
    system_access_key: P9Eb6S8XNyo4dtZZUUMy
    system_secret_key: qqHCUtfdNnpHq3PZRHW5un9l0bEBM812Uhow0XfB
    radosgw_address: "{{ _radosgw_address }}"
    radosgw_frontend_port: 8080
```

f. **可选：要创建多个实例，编辑该文件并将配置详情添加到对应对象网关节点上的所有实例。配置以下设置，并更新 rgw_instances 下的项目。将 multi-site-keys-realm-1.txt 文件中保存的随机字符串用于 *ACCESS_KEY_1* 和 *SECRET_KEY_1*。**

语法

```
rgw_instances:
  - instance_name: 'INSTANCE_NAME_1'
```

```
    rgw_multisite: true
    rgw_zonemaster: true
    rgw_zonesecondary: false
    rgw_zonegroupmaster: true
    rgw_zone: ZONE_NAME_1
    rgw_zonegroup: ZONE_GROUP_NAME_1
    rgw_realm: REALM_NAME_1
    rgw_zone_user: ZONE_USER_NAME_1
    rgw_zone_user_display_name: "ZONE_DISPLAY_NAME_1"
    system_access_key: ACCESS_KEY_1
    system_secret_key: SECRET_KEY_1
    radosgw_address: "{{ _radosgw_address }}"
    radosgw_frontend_port: PORT_NUMBER_1
  - instance_name: 'INSTANCE_NAME_2'
    rgw_multisite: true
    rgw_zonemaster: true
    rgw_zonesecondary: false
    rgw_zonegroupmaster: true
    rgw_zone: ZONE_NAME_1
    rgw_zonegroup: ZONE_GROUP_NAME_1
    rgw_realm: REALM_NAME_1
    rgw_zone_user: ZONE_USER_NAME_1
    rgw_zone_user_display_name: "ZONE_DISPLAY_NAME_1"
    system_access_key: ACCESS_KEY_1
    system_secret_key: SECRET_KEY_1
    radosgw_address: "{{ _radosgw_address }}"
    radosgw_frontend_port: PORT_NUMBER_2
```

示例

```
rgw_instances:
  - instance_name: 'rgw0'
    rgw_multisite: true
    rgw_zonemaster: true
    rgw_zonesecondary: false
    rgw_zonegroupmaster: true
    rgw_zone: paris
    rgw_zonegroup: idf
    rgw_realm: france
    rgw_zone_user: jacques.chirac
    rgw_zone_user_display_name: "Jacques Chirac"
    system_access_key: P9Eb6S8XNyo4dtZZUUMy
    system_secret_key: qqHCUtfdNnpHq3PZRHW5un9l0bEBM812Uhow0XfB
    radosgw_address: "{{ _radosgw_address }}"
    radosgw_frontend_port: 8080
  - instance_name: 'rgw1'
    rgw_multisite: true
    rgw_zonemaster: true
    rgw_zonesecondary: false
```

```
rgw_zonegroupmaster: true
rgw_zone: paris
rgw_zonegroup: idf
rgw_realm: france
rgw_zone_user: jacques.chirac
rgw_zone_user_display_name: "Jacques Chirac"
system_access_key: P9Eb6S8XNyo4dtZZUUMy
system_secret_key: qqHCUtfdNnpHq3PZRHW5un9l0bEBM812Uhow0XfB
radosgw_address: "{{ _radosgw_address }}"
radosgw_frontend_port: 8081
```

辅助存储集群

a. 进入 **Ceph-ansible 配置目录：**

```
[root@ansible ~]# cd /usr/share/ceph-ansible
```

b. **打开并编辑 group_vars/all.yml 文件。取消注释 rgw_multisite 行并将其设置为 true。取消注释 rgw_multisite_proto 参数。**

```
rgw_multisite: true
rgw_multisite_proto: "http"
```

c. **在 /usr/share/ceph-ansible 中创建 host_vars 目录：**

```
[root@ansible ceph-ansible]# mkdir host_vars
```

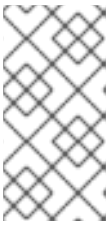
d. **在 host_vars 中为次要存储集群上的每个对象网关节点创建一个文件。文件名应当与 Ansible 清单文件中使用的名称相同。例如，如果对象网关节点命名为 rgw-secondary，则创建 host_vars/rgw-secondary 文件。**

语法

```
touch host_vars/NODE_NAME
```

示例

```
[root@ansible ceph-ansible]# touch host_vars/rgw-secondary
```



注意

如果集群中有多个 Ceph 对象网关节点用于多站点配置，则为每个节点创建文件。

e.

配置以下设置：使用与 *ZONE_USER_NAME*、*ZONE_DISPLAY_NAME*、*ACCESS_KEY*、*SECRET_KEY*、*REALM_NAME* 和 *ZONE_GROUP_NAME* 相同的值。为主存储集群中的 *ZONE_NAME* 使用不同的值。将 *MASTER_RGW_NODE_NAME* 设置为 master 区域的 Ceph 对象网关节点。请注意，与主存储集群相比，rgw_zonemaster、rgw_zonesecondary 和 rgw_zonegroupmaster 的设置将被撤销。

语法

```
rgw_instances:
  - instance_name: 'INSTANCE_NAME_1'
      rgw_multisite: true
      rgw_zonemaster: false
      rgw_zonesecondary: true
      rgw_zonegroupmaster: false
      rgw_zone: ZONE_NAME_2
      rgw_zonegroup: ZONE_GROUP_NAME_1
      rgw_realm: REALM_NAME_1
      rgw_zone_user: ZONE_USER_NAME_1
      rgw_zone_user_display_name: "ZONE_DISPLAY_NAME_1"
      system_access_key: ACCESS_KEY_1
      system_secret_key: SECRET_KEY_1
      radosgw_address: "{{ _radosgw_address }}"
      radosgw_frontend_port: PORT_NUMBER_1
      endpoint:
RGW_PRIMARY_HOSTNAME_ENDPOINT:RGW_PRIMARY_PORT_NUMBER_1
```

示例

```
rgw_instances:
  - instance_name: 'rgw0'
      rgw_multisite: true
      rgw_zonemaster: false
      rgw_zonesecondary: true
      rgw_zonegroupmaster: false
      rgw_zone: lyon
      rgw_zonegroup: idf
      rgw_realm: france
      rgw_zone_user: jacques.chirac
      rgw_zone_user_display_name: "Jacques Chirac"
      system_access_key: P9Eb6S8XNyo4dtZZUUMy
      system_secret_key: qqHCUtfdNnpHq3PZRHW5un9l0bEBM812Uhow0XfB
      radosgw_address: "{{ _radosgw_address }}"
      radosgw_frontend_port: 8080
      endpoint: http://rgw-primary:8081
```

f.

可选：要创建多个实例，编辑该文件并将配置详情添加到对应对象网关节点上的所有实例。配置以下设置，并更新 rgw_instances 下的项目。将 multi-site-keys-realm-1.txt 文件中保存的随机字符串用于 *ACCESS_KEY_1* 和 *SECRET_KEY_1*。将 *RGW_PRIMARY_HOSTNAME* 设置为主存储集群中的对象网关节点。

语法

```
rgw_instances:
  - instance_name: 'INSTANCE_NAME_1'
      rgw_multisite: true
      rgw_zonemaster: false
      rgw_zonesecondary: true
      rgw_zonegroupmaster: false
      rgw_zone: ZONE_NAME_2
      rgw_zonegroup: ZONE_GROUP_NAME_1
      rgw_realm: REALM_NAME_1
      rgw_zone_user: ZONE_USER_NAME_1
      rgw_zone_user_display_name: "ZONE_DISPLAY_NAME_1"
      system_access_key: ACCESS_KEY_1
      system_secret_key: SECRET_KEY_1
      radosgw_address: "{{ _radosgw_address }}"
      radosgw_frontend_port: PORT_NUMBER_1
      endpoint: RGW_PRIMARY_HOSTNAME:RGW_PRIMARY_PORT_NUMBER_1
  - instance_name: '_INSTANCE_NAME_2_'
      rgw_multisite: true
      rgw_zonemaster: false
      rgw_zonesecondary: true
```

```
    rgw_zonegroupmaster: false
    rgw_zone: ZONE_NAME_2
    rgw_zonegroup: ZONE_GROUP_NAME_1
    rgw_realm: REALM_NAME_1
    rgw_zone_user: ZONE_USER_NAME_1
    rgw_zone_user_display_name: "ZONE_DISPLAY_NAME_1"
    system_access_key: ACCESS_KEY_1
    system_secret_key: SECRET_KEY_1
    radosgw_address: "{{ _radosgw_address }}"
    radosgw_frontend_port:  PORT_NUMBER_1
    endpoint: RGW_PRIMARY_HOSTNAME:RGW_PRIMARY_PORT_NUMBER_2
```

示例

```
rgw_instances:
  - instance_name: 'rgw0'
      rgw_multisite: true
      rgw_zonemaster: false
      rgw_zonesecondary: true
      rgw_zonegroupmaster: false
      rgw_zone: lyon
      rgw_zonegroup: idf
      rgw_realm: france
      rgw_zone_user: jacques.chirac
      rgw_zone_user_display_name: "Jacques Chirac"
      system_access_key: P9Eb6S8XNyo4dtZZUUMy
      system_secret_key: qqHCUtfdNnpHq3PZRHW5un9l0bEBM812Uhow0XfB
      radosgw_address: "{{ _radosgw_address }}"
      radosgw_frontend_port: 8080
      endpoint: http://rgw-primary:8080
  - instance_name: 'rgw1'
      rgw_multisite: true
      rgw_zonemaster: false
      rgw_zonesecondary: true
      rgw_zonegroupmaster: false
      rgw_zone: lyon
      rgw_zonegroup: idf
      rgw_realm: france
      rgw_zone_user: jacques.chirac
      rgw_zone_user_display_name: "Jacques Chirac"
      system_access_key: P9Eb6S8XNyo4dtZZUUMy
      system_secret_key: qqHCUtfdNnpHq3PZRHW5un9l0bEBM812Uhow0XfB
      radosgw_address: "{{ _radosgw_address }}"
      radosgw_frontend_port: 8081
      endpoint: http://rgw-primary:8081
```

在两个站点中，执行以下步骤：

1. 在主存储集群上运行 Ansible playbook：

   - 裸机部署：

     ```
     [user@ansible ceph-ansible]$ ansible-playbook site.yml -i hosts
     ```

   - 容器部署：

     ```
     [user@ansible ceph-ansible]$ ansible-playbook site-container.yml -i hosts
     ```

2. 验证辅助存储集群可以访问主存储集群中的 API。

   在辅助存储集群中的 Object Gateway 节点中，使用 curl 或者另一个 HTTP 客户端连接到主集群中的 API。使用用于在 all.yml 中配置 rgw_pull_proto、rgw_pullhost 和 rgw_pull_port 的信息编写 URL。在上例中，URL 是 http://cluster0-rgw-000:8080。如果无法访问 API，请验证 URL 是否正确，并根据需要更新 all.yml。URL 正常工作并解决所有网络问题后，请继续下一步，以在次要存储集群上运行 Ansible playbook。

3. 在辅助存储集群上运行 Ansible playbook:

   注意

   如果部署了集群，且您只对 Ceph 对象网关进行了更改，则使用 --limit rgws 选项。

   - 裸机部署：

     ```
     [user@ansible ceph-ansible]$ ansible-playbook site.yml -i hosts
     ```

   - 容器部署：

     ```
     [user@ansible ceph-ansible]$ ansible-playbook site-container.yml -i hosts
     ```

   在主存储和次要存储集群上运行 Ansible playbook 后，Ceph 对象网关以主动-主动状

态运行。

4. 验证两个站点上的多站点 Ceph 对象网关配置：

语法

```
radosgw-admin sync status
```

5.7.3. 使用多个域和多个实例配置多站点 Ceph 对象网关

Ceph-ansible 配置 Ceph 对象网关，以在具有多个 Ceph 对象网关实例的多个存储集群之间镜像数据。

警告

在单一站点配置中已使用网关后，Ceph-ansible 无法将网关重新配置为多站点设置。您可以手动部署此配置。联系*红帽支持*以获取帮助。

先决条件

- 两个正在运行的 Red Hat Ceph Storage 集群。
- 每个存储集群中至少有两个对象网关节点。
- 在 Ceph 对象网关节点上，执行 *Red Hat Ceph Storage安装指南*中的*安装 Red Hat Ceph Storage 要求*一节中列出的任务。
- 对于每个对象网关节点，执行 *Red Hat Ceph Storage 安装指南*中的*安装 Ceph 对象网关*一节中的第 1 到 6 步。

流程

1. **在任何节点上，为 realm 1 和 2 生成系统访问密钥和密钥，并将它们分别保存在名为 multi-site-keys-realm-1.txt 和 multi-site-keys-realm-2.txt 的文件中：**

```
# echo system_access_key: $(cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 20 | head -n 1) > multi-site-keys-realm-1.txt
[root@ansible ~]# echo system_secret_key: $(cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 40 | head -n 1) >> multi-site-keys-realm-1.txt

# echo system_access_key: $(cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 20 | head -n 1) > multi-site-keys-realm-2.txt
[root@ansible ~]# echo system_secret_key: $(cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 40 | head -n 1) >> multi-site-keys-realm-2.txt
```

site-A 存储集群

a. **进入 Ansible 配置目录：**

```
[root@ansible ~]# cd /usr/share/ceph-ansible
```

b. **打开并编辑 group_vars/all.yml 文件。取消注释 rgw_multisite 行并将其设置为 true。取消注释 rgw_multisite_proto 参数。**

```
rgw_multisite: true
rgw_multisite_proto: "http"
```

c. **在 /usr/share/ceph-ansible 中创建 host_vars 目录：**

```
[root@ansible ceph-ansible]# mkdir host_vars
```

d. **在 host_vars 中为 site-A 存储集群上的每个对象网关节点创建一个文件。文件名应当与 Ansible 清单文件中使用的名称相同。例如，如果对象网关节点命名为 rgw-site-a，则创建 host_vars/rgw-site-a 文件。**

语法

```
touch host_vars/NODE_NAME
```

示例

```
[root@ansible ceph-ansible]# touch host_vars/rgw-site-a
```

注意

如果集群中有多个 Ceph 对象网关节点用于多站点配置，则为每个节点创建单独的文件。

e.

要为第一个域创建多个实例，请编辑文件，并将配置详情添加到对应对象网关节点上的所有实例。配置以下设置，以及更新第一个域的 rgw_instances 下的项目。将 multi-site-keys-realm-1.txt 文件中保存的随机字符串用于 *ACCESS_KEY_1* 和 *SECRET_KEY_1*。

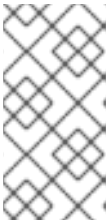
语法

```
rgw_instances:
  - instance_name: '_INSTANCE_NAME_1_'
      rgw_multisite: true
      rgw_zonemaster: true
      rgw_zonesecondary: false
      rgw_zonegroupmaster: true
      rgw_zone: ZONE_NAME_1
      rgw_zonegroup: ZONE_GROUP_NAME_1
      rgw_realm: REALM_NAME_1
      rgw_zone_user: ZONE_USER_NAME_1
      rgw_zone_user_display_name: "ZONE_DISPLAY_NAME_1"
      system_access_key: ACCESS_KEY_1
      system_secret_key: SECRET_KEY_1
      radosgw_address: "{{ _radosgw_address }}"
      radosgw_frontend_port: PORT_NUMBER_1
    - instance_name: '_INSTANCE_NAME_2_'
      rgw_multisite: true
      rgw_zonemaster: true
      rgw_zonesecondary: false
      rgw_zonegroupmaster: true
      rgw_zone: ZONE_NAME_1
      rgw_zonegroup: ZONE_GROUP_NAME_1
      rgw_realm: REALM_NAME_1
```

```
    rgw_zone_user: ZONE_USER_NAME_1
    rgw_zone_user_display_name: "ZONE_DISPLAY_NAME_1"
    system_access_key: ACCESS_KEY_1
    system_secret_key: SECRET_KEY_1
    radosgw_address: "{{ _radosgw_address }}"
    radosgw_frontend_port: PORT_NUMBER_1
```

示例

```
rgw_instances:
  - instance_name: 'rgw0'
      rgw_multisite: true
      rgw_zonemaster: true
      rgw_zonesecondary: false
      rgw_zonegroupmaster: true
      rgw_zone: paris
      rgw_zonegroup: idf
      rgw_realm: france
      rgw_zone_user: jacques.chirac
      rgw_zone_user_display_name: "Jacques Chirac"
      system_access_key: P9Eb6S8XNyo4dtZZUUMy
      system_secret_key: qqHCUtfdNnpHq3PZRHW5un9l0bEBM812Uhow0XfB
      radosgw_address: "{{ _radosgw_address }}"
      radosgw_frontend_port: 8080
  - instance_name: 'rgw1'
      rgw_multisite: true
      rgw_zonemaster: true
      rgw_zonesecondary: false
      rgw_zonegroupmaster: true
      rgw_zone: paris
      rgw_zonegroup: idf
      rgw_realm: france
      rgw_zone_user: jacques.chirac
      rgw_zone_user_display_name: "Jacques Chirac"
      system_access_key: P9Eb6S8XNyo4dtZZUUMy
      system_secret_key: qqHCUtfdNnpHq3PZRHW5un9l0bEBM812Uhow0XfB
      radosgw_address: "{{ _radosgw_address }}"
      radosgw_frontend_port: 8080
```



注意

在将 site-B 上的所有域配置为 site-A 为次要域后，跳过下一步并运行它，然后运行 Ansible playbook。

f.

对于其他域的多个实例，请配置以下设置，以及更新 rgw_instances 下的项目。使用 multi-site-keys-realm-2.txt 文件中保存的随机字符串用于 *ACCESS_KEY_2* 和 *SECRET_KEY_2*。

语法

```
rgw_instances:
  - instance_name: 'INSTANCE_NAME_1'
      rgw_multisite: true
      rgw_zonemaster: false
      rgw_zonesecondary: true
      rgw_zonegroupmaster: false
      rgw_zone: ZONE_NAME_2
      rgw_zonegroup: ZONE_GROUP_NAME_2
      rgw_realm: REALM_NAME_2
      rgw_zone_user: ZONE_USER_NAME_2
      rgw_zone_user_display_name: "ZONE_DISPLAY_NAME_2"
      system_access_key: ACCESS_KEY_2
      system_secret_key: SECRET_KEY_2
      radosgw_address: "{{ _radosgw_address }}"
      radosgw_frontend_port: PORT_NUMBER_1
      endpoint:
RGW_SITE_B_PRIMARY_HOSTNAME_ENDPOINT:RGW_SITE_B_PORT_NUMBER_1
  - instance_name: 'INSTANCE_NAME_2'
      rgw_multisite: true
      rgw_zonemaster: false
      rgw_zonesecondary: true
      rgw_zonegroupmaster: false
      rgw_zone: ZONE_NAME_2
      rgw_zonegroup: ZONE_GROUP_NAME_2
      rgw_realm: REALM_NAME_2
      rgw_zone_user: ZONE_USER_NAME_2
      rgw_zone_user_display_name: "ZONE_DISPLAY_NAME_2"
      system_access_key: ACCESS_KEY_2
      system_secret_key: SECRET_KEY_2
      radosgw_address: "{{ _radosgw_address }}"
      radosgw_frontend_port: PORT_NUMBER_1
      endpoint:
RGW_SITE_B_PRIMARY_HOSTNAME_ENDPOINT:RGW_SITE_B_PORT_NUMBER_1
```

示例

```
rgw_instances:
  - instance_name: 'rgw0'
      rgw_multisite: true
      rgw_zonemaster: false
      rgw_zonesecondary: true
      rgw_zonegroupmaster: false
      rgw_zone: fairbanks
      rgw_zonegroup: alaska
      rgw_realm: usa
      rgw_zone_user: edward.lewis
      rgw_zone_user_display_name: "Edward Lewis"
      system_access_key: yu17wkvAx3B8Wyn08XoF
      system_secret_key: 5YZfaSUPqxSNIkZQQA3lBZ495hnIV6k2HAz710BY
      radosgw_address: "{{ _radosgw_address }}"
      radosgw_frontend_port: 8080
      endpoint: http://rgw-site-b:8081
  - instance_name: 'rgw1'
      rgw_multisite: true
      rgw_zonemaster: false
      rgw_zonesecondary: true
      rgw_zonegroupmaster: false
      rgw_zone: fairbanks
      rgw_zonegroup: alaska
      rgw_realm: usa
      rgw_zone_user: edward.lewis
      rgw_zone_user_display_name: "Edward Lewis"
      system_access_key: yu17wkvAx3B8Wyn08XoF
      system_secret_key: 5YZfaSUPqxSNIkZQQA3lBZ495hnIV6k2HAz710BY
      radosgw_address: "{{ _radosgw_address }}"
      radosgw_frontend_port: 8081
      endpoint: http://rgw-site-b:8081
```

g. **在 site-A 存储集群上运行 Ansible playbook:**

- **裸机部署：**

```
[user@ansible ceph-ansible]$ ansible-playbook site.yml -i hosts
```

- **容器部署：**

```
[user@ansible ceph-ansible]$ ansible-playbook site-container.yml -i hosts
```

Site-B Storage Cluster

a. 进入 **Ceph-ansible** 配置目录：

```
[root@ansible ~]# cd /usr/share/ceph-ansible
```

b. 打开并编辑 **group_vars/all.yml** 文件。取消注释 **rgw_multisite** 行并将其设置为 **true**。取消注释 **rgw_multisite_proto** 参数。

```
rgw_multisite: true
rgw_multisite_proto: "http"
```

c. 在 **/usr/share/ceph-ansible** 中创建 **host_vars** 目录：

```
[root@ansible ceph-ansible]# mkdir host_vars
```

d. 在 **host_vars** 中为 **site-B** 存储集群上的每个对象网关节点创建一个文件。文件名应当与 **Ansible** 清单文件中使用的名称相同。例如，如果对象网关节点命名为 **rgw-site-b**，则创建 **host_vars/rgw-site-b** 文件。

语法

```
touch host_vars/NODE_NAME
```

示例

```
[root@ansible ceph-ansible]# touch host_vars/rgw-site-b
```

注意

如果集群中有多个 Ceph 对象网关节点用于多站点配置，则为每个节点创建文件。

e.

要为第一个域创建多个实例，请编辑文件，并将配置详情添加到对应对象网关节点上的所有实例。配置以下设置，以及更新第一个域的 rgw_instances 下的项目。将 multi-site-keys-realm-1.txt 文件中保存的随机字符串用于 *ACCESS_KEY_1* 和 *SECRET_KEY_1*。将 *RGW_SITE_A_PRIMARY_HOSTNAME_ENDPOINT* 设置为 site-A 存储集群中的对象网关节点。

语法

```
rgw_instances:
  - instance_name: '_INSTANCE_NAME_1_'
      rgw_multisite: true
      rgw_zonemaster: false
      rgw_zonesecondary: true
      rgw_zonegroupmaster: false
      rgw_zone: ZONE_NAME_1
      rgw_zonegroup: ZONE_GROUP_NAME_1
      rgw_realm: REALM_NAME_1
      rgw_zone_user: ZONE_USER_NAME_1
      rgw_zone_user_display_name: "ZONE_DISPLAY_NAME_1"
      system_access_key: ACCESS_KEY_1
      system_secret_key: SECRET_KEY_1
      radosgw_address: "{{ _radosgw_address }}"
      radosgw_frontend_port: PORT_NUMBER_1
      endpoint: RGW_SITE_A_HOSTNAME_ENDPOINT:RGW_SITE_A_PORT_NUMBER_1
  - instance_name: '_INSTANCE_NAME_2_'
      rgw_multisite: true
      rgw_zonemaster: false
      rgw_zonesecondary: true
      rgw_zonegroupmaster: false
      rgw_zone: ZONE_NAME_1
      rgw_zonegroup: ZONE_GROUP_NAME_1
      rgw_realm: REALM_NAME_1
      rgw_zone_user: ZONE_USER_NAME_1
      rgw_zone_user_display_name: "ZONE_DISPLAY_NAME_1"
      system_access_key: ACCESS_KEY_1
      system_secret_key: SECRET_KEY_1
      radosgw_address: "{{ _radosgw_address }}"
      radosgw_frontend_port:  PORT_NUMBER_1
      endpoint:
RGW_SITE_A_PRIMARY_HOSTNAME_ENDPOINT:RGW_SITE_A_PORT_NUMBER_1
```

示例

```
rgw_instances:
  - instance_name: 'rgw0'
      rgw_multisite: true
      rgw_zonemaster: false
      rgw_zonesecondary: true
      rgw_zonegroupmaster: false
      rgw_zone: paris
      rgw_zonegroup: idf
      rgw_realm: france
      rgw_zone_user: jacques.chirac
      rgw_zone_user_display_name: "Jacques Chirac"
      system_access_key: P9Eb6S8XNyo4dtZZUUMy
      system_secret_key: qqHCUtfdNnpHq3PZRHW5un9l0bEBM812Uhow0XfB
      radosgw_address: "{{ _radosgw_address }}"
      radosgw_frontend_port: 8080
      endpoint: http://rgw-site-a:8080
  - instance_name: 'rgw1'
      rgw_multisite: true
      rgw_zonemaster: false
      rgw_zonesecondary: true
      rgw_zonegroupmaster: false
      rgw_zone: paris
      rgw_zonegroup: idf
      rgw_realm: france
      rgw_zone_user: jacques.chirac
      rgw_zone_user_display_name: "Jacques Chirac"
      system_access_key: P9Eb6S8XNyo4dtZZUUMy
      system_secret_key: qqHCUtfdNnpHq3PZRHW5un9l0bEBM812Uhow0XfB
      radosgw_address: "{{ _radosgw_address }}"
      radosgw_frontend_port: 8081
      endpoint: http://rgw-site-a:8081
```

f.

对于其他域的多个实例，请配置以下设置，以及更新 **rgw_instances** 下的项目。使用 **multi-site-keys-realm-2.txt** 文件中保存的随机字符串用于 ***ACCESS_KEY_2*** 和 ***SECRET_KEY_2***。将 ***RGW_SITE_A_PRIMARY_HOSTNAME_ENDPOINT*** 设置为 **site-A** 存储集群中的对象网关节点。

语法

```
rgw_instances:
  - instance_name: 'INSTANCE_NAME_1'
      rgw_multisite: true
```

```
    rgw_zonemaster: true
    rgw_zonesecondary: false
    rgw_zonegroupmaster: true
    rgw_zone: ZONE_NAME_2
    rgw_zonegroup: ZONE_GROUP_NAME_2
    rgw_realm: REALM_NAME_2
    rgw_zone_user: ZONE_USER_NAME_2
    rgw_zone_user_display_name: "ZONE_DISPLAY_NAME_2"
    system_access_key: ACCESS_KEY_2
    system_secret_key: SECRET_KEY_2
    radosgw_address: "{{ _radosgw_address }}"
    radosgw_frontend_port: PORT_NUMBER_1
  - instance_name: '_INSTANCE_NAME_2_'
    rgw_multisite: true
    rgw_zonemaster: true
    rgw_zonesecondary: false
    rgw_zonegroupmaster: true
    rgw_zone: ZONE_NAME_2
    rgw_zonegroup: ZONE_GROUP_NAME_2
    rgw_realm: REALM_NAME_2
    rgw_zone_user: ZONE_USER_NAME_2
    rgw_zone_user_display_name: "ZONE_DISPLAY_NAME_2"
    system_access_key: ACCESS_KEY_2
    system_secret_key: SECRET_KEY_2
    radosgw_address: "{{ _radosgw_address }}"
    radosgw_frontend_port: PORT_NUMBER_1
```

示例

```
rgw_instances:
  - instance_name: 'rgw0'
    rgw_multisite: true
    rgw_zonemaster: true
    rgw_zonesecondary: false
    rgw_zonegroupmaster: true
    rgw_zone: fairbanks
    rgw_zonegroup: alaska
    rgw_realm: usa
    rgw_zone_user: edward.lewis
    rgw_zone_user_display_name: "Edward Lewis"
    system_access_key: yu17wkvAx3B8Wyn08XoF
    system_secret_key: 5YZfaSUPqxSNIkZQQA3IBZ495hnIV6k2HAz710BY
    radosgw_address: "{{ _radosgw_address }}"
    radosgw_frontend_port: 8080
  - instance_name: 'rgw1'
    rgw_multisite: true
    rgw_zonemaster: true
    rgw_zonesecondary: false
    rgw_zonegroupmaster: true
```

```
rgw_zone: fairbanks
rgw_zonegroup: alaska
rgw_realm: usa
rgw_zone_user: edward.lewis
rgw_zone_user_display_name: "Edward Lewis"
system_access_key: yu17wkvAx3B8Wyn08XoF
system_secret_key: 5YZfaSUPqxSNlkZQQA3lBZ495hnIV6k2HAz710BY
radosgw_address: "{{ _radosgw_address }}"
radosgw_frontend_port: 8081
```

g. **在 site-B 存储集群上运行 Ansible playbook:**

- **裸机部署：**

```
[user@ansible ceph-ansible]$ ansible-playbook site.yml -i hosts
```

- **容器部署：**

```
[user@ansible ceph-ansible]$ ansible-playbook site-container.yml -i hosts
```

在 site-A 存储集群上针对其他 site-A 的域再次运行 Ansible playbook。

在 site-A 和 site-B 存储集群上运行 Ansible playbook 后，Ceph 对象网关以主动-主动状态运行。

验证

1. 验证多站点 **Ceph** 对象网关配置：

   a. **从每个站点的 Ceph monitor 和对象网关节点(site-A 和 site-B)，使用 curl 或其他 HTTP 客户端来验证是否可从其他站点访问 API。**

   b. 对两个站点运行 **radosgw-admin sync status** 命令。

   语法

```
radosgw-admin sync status
radosgw-admin sync status --rgw -realm REALM_NAME (1)
```

(1) 对存储集群的对应节点上的多个域使用这个选项。

示例

```
[user@ansible ceph-ansible]$ radosgw-admin sync status

[user@ansible ceph-ansible]$ radosgw-admin sync status --rgw -realm usa
```

5.8. 在同一主机上部署具有不同硬件的 OSD

您可以使用 Ansible 中的 device_class 功能将混合 OSD（如 HDD 和 SSD）部署到同一主机上。

先决条件

- 有效的客户订阅。
- 对 Ansible 管理节点的根级别访问权限。
- 启用 Red Hat Ceph Storage 工具和 Ansible 存储库。
- 用于 Ansible 应用的 ansible 用户帐户。

- 已部署 OSD。

流程

1. 在 group_vars/mons.yml 文件中创建 crush_rules:

示例

```
crush_rule_config: true
crush_rule_hdd:
   name: HDD
   root: default
   type: host
   class: hdd
   default: true
crush_rule_ssd:
   name: SSD
   root: default
   type: host
   class: ssd
   default: true
crush_rules:
     - "{{ crush_rule_hdd }}"
     - "{{ crush_rule_ssd }}"
create_crush_tree: true
```

注意

如果您在集群中没有使用 SSD 或 HDD 设备，请不要为该设备定义 crush_rules。

2. 使用在 group_vars/clients.yml 文件中创建的 crush_rules 来创建 pools。

示例

```
copy_admin_key: True
user_config: True
```

```
pool1:
  name: "pool1"
  pg_num: 128
  pgp_num: 128
  rule_name: "HDD"
  type: "replicated"
  device_class: "hdd"
pools:
  - "{{ pool1 }}"
```

3. **将 roots 分配给 OSD 的清单文件示例：**

示例

```
[mons]
mon1

[osds]
osd1 osd_crush_location="{ 'root': 'default', 'rack': 'rack1', 'host': 'osd1' }"
osd2 osd_crush_location="{ 'root': 'default', 'rack': 'rack1', 'host': 'osd2' }"
osd3 osd_crush_location="{ 'root': 'default', 'rack': 'rack2', 'host': 'osd3' }"
osd4 osd_crush_location="{ 'root': 'default', 'rack': 'rack2', 'host': 'osd4' }"
osd5 devices="['/dev/sda', '/dev/sdb']" osd_crush_location="{ 'root': 'default', 'rack': 'rack3',
'host': 'osd5' }"
osd6 devices="['/dev/sda', '/dev/sdb']" osd_crush_location="{ 'root': 'default', 'rack': 'rack3',
'host': 'osd6' }"

[mgrs]
mgr1

[clients]
client1
```

4. **查看树。**

语法

```
[root@mon ~]# ceph osd tree
```

示例

```
TYPE NAME

root default
    rack rack1
      host osd1
          osd.0
          osd.10
      host osd2
          osd.3
          osd.7
          osd.12
    rack rack2
      host osd3
          osd.1
          osd.6
          osd.11
      host osd4
          osd.4
          osd.9
          osd.13
    rack rack3
       host osd5
          osd.2
          osd.8
       host osd6
          osd.14
          osd.15
```

5. **验证池。**

示例

```
# for i in $(rados lspools);do echo "pool: $i"; ceph osd pool get $i crush_rule;done

pool: pool1
crush_rule: HDD
```

其它资源

- 如需了解更多详细信息，请参阅 *Red Hat Ceph Storage 安装指南*中的安装 *Red Hat Ceph Storage 集群*。

- 如需了解更多详细信息，请参阅 *Red Hat Ceph Storage策略指南*中的*设备类*。

5.9. 安装 NFS-GANESHA 网关

Ceph NFS Ganesha 网关是在 Ceph 对象网关基础上构建的 NFS 接口，为应用提供 POSIX 文件系统接口到 Ceph 对象网关，以便在文件系统内将文件迁移到 Ceph 对象存储。

先决条件

- 正在运行的 Ceph 存储集群，最好处于 active + clean 状态。

- 至少一个运行 Ceph 对象网关的节点.

- 在尝试运行 NFS-Ganesha 之前，禁用将运行 NFS-Ganesha 的任何主机上运行的任何内核 NFS 服务实例。如果另一个 NFS 实例正在运行，NFS-Ganesha 将不会启动。

*启用免密码 SSH 访问。

- 确保 rpcbind 服务正在运行：

```
# systemctl start rpcbind
```

注意

默认情况下，通常安装提供 rpcbind 的 rpcbind 软件包。否则，请先安装软件包。

- 如果 **nfs-service** 服务正在运行，请停止并禁用该服务：

```
# systemctl stop nfs-server.service
# systemctl disable nfs-server.service
```

流程

在 **Ansible** 管理节点上执行下列任务：

1. 从示例文件创建 **nfss.yml** 文件：

```
[root@ansible ~]# cd /usr/share/ceph-ansible/group_vars
[root@ansible ~]# cp nfss.yml.sample nfss.yml
```

2. 将网关主机添加到 **[nfss]** 组下的 **/etc/ansible/hosts** 文件中，以识别其组成员资格。

```
[nfss]
NFS_HOST_NAME_1
NFS_HOST_NAME_2
NFS_HOST_NAME[3..10]
```

如果主机具有连续命名，则可以使用范围指定符，例如：**[3..10]**。

3. 进入 **Ansible** 配置目录：

```
[root@ansible ~]# cd /usr/share/ceph-ansible
```

4. 要将管理员密钥复制到 **Ceph** 对象网关节点，请取消注释 **/usr/share/ceph-ansible/group_vars/nfss.yml** 文件中的 **copy_admin_key** 设置：

```
copy_admin_key: true
```

5. 配置 **/usr/share/ceph-ansible/group_vars/nfss.yml** 文件的 **FSAL (File System Abstraction Layer)** 部分。提供导出 **ID (*NUMERIC_EXPORT_ID*)**、**S3** 用户 **ID (*S3_USER*)**、**S3** 访问密钥 **(*ACCESS_KEY*)** 和密钥密钥 **(*SECRET_KEY*)**：

```
# FSAL RGW Config #
```

```
ceph_nfs_rgw_export_id: NUMERIC_EXPORT_ID
#ceph_nfs_rgw_pseudo_path: "/"
#ceph_nfs_rgw_protocols: "3,4"
#ceph_nfs_rgw_access_type: "RW"
ceph_nfs_rgw_user: "S3_USER"
ceph_nfs_rgw_access_key: "ACCESS_KEY"
ceph_nfs_rgw_secret_key: "SECRET_KEY"
```

警告

访问和密钥是可选的，可以生成。

6. 运行 **Ansible playbook:**

   a. **裸机部署：**

   ```
   [ansible@admin ceph-ansible]$ ansible-playbook site.yml --limit nfss -i hosts
   ```

   b. **容器部署：**

   ```
   [ansible@admin ceph-ansible]$ ansible-playbook site-container.yml --limit nfss -i hosts
   ```

其它资源

- *了解 限制 选项*
- *对象网关配置和管理指南*

5.10. 了解 LIMIT 选项

本节包含有关 **Ansible --limit** 选项的信息。

Ansible 支持 --limit 选项，允许您将 site 和 site-container Ansible playbook 用于清单文件的特定角色。

```
ansible-playbook site.yml|site-container.yml --limit osds|rgws|clients|mdss|nfss|iscsigws -i hosts
```

裸机

例如，若要仅在裸机上重新部署 OSD，请以 Ansible 用户身份运行以下命令：

```
[ansible@ansible ceph-ansible]$ ansible-playbook site.yml --limit osds -i hosts
```

容器

例如，若要仅重新部署容器上的 OSD，请以 Ansible 用户身份运行以下命令：

```
[ansible@ansible ceph-ansible]$ ansible-playbook site-container.yml --limit osds -i hosts
```

5.11. 放置组自动扩展

PG 调优使用 PG 计算器手动插入 pg_num 的数字。从 Red Hat Ceph Storage 4.1 开始，可以通过启用 pg_autoscaler Ceph Manager 模块来自动进行 PG 调优。PG 自动缩放器以每个池为基础配置，且以 2 的电源扩展 pg_num。如果建议的值是实际值的三倍以上，PG 自动缩放器才会提议更改 pg_num。

PG 自动缩放器具有三种模式：

warn

新池和现有池的默认模式。如果建议的 pg_num 值与当前的 pg_num 值有太大差别，则会生成健康警告。

on

池的 pg_num 会自动调整。

off

自动缩放器可以针对任何池进行关闭，但存储管理员需要手动为池设置 pg_num 值。

一旦为池启用了 PG 自动缩放器，您可以通过运行 ceph osd pool autoscale-status 命令来查看值调整。autoscale-status 命令显示池的当前状态。以下是 autoscale-status 列描述：

SIZE

报告池中存储的数据总量，以字节为单位。这个大小包括对象数据和 OMAP 数据。

TARGET SIZE

报告存储管理员提供的池的预期大小。此值用于计算池的理想 PG 数量。

RATE

复制 bucket 的复制因子，或纠删代码池的比例。

RAW CAPACITY

池映射到基于 CRUSH 的存储设备的原始存储容量。

RATIO

池消耗的存储总数的比率。

TARGET RATIO

一个比率，用于指定存储集群总空间中由存储管理员提供的池消耗的占比。

PG_NUM

池的当前 PG 数量。

NEW PG_NUM

建议的值。可能没有设置这个值。

AUTOSCALE

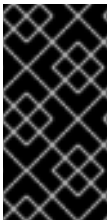
为池设置的 PG 自动缩放器模式。

其它资源

- 放置组池计算器。

5.11.1. 配置放置组自动扩展

您可以配置 Ceph Ansible，以便为 Red Hat Ceph Storage 集群中的新池启用和配置 PG 自动缩放器。默认情况下，放置组 (PG) 自动缩放器处于 off 状态。



重要

目前，您只能在新的 Red Hat Ceph Storage 部署中配置放置组自动扩展器，而不能在现有 Red Hat Ceph Storage 安装中配置。

先决条件

- 访问 **Ansible 管理节点.**
- 访问 **Ceph 监控节点.**

流程

1. **在 Ansible 管理节点上，打开 group_vars/all.yml 文件进行编辑。**
2. **将 pg_autoscale_mode 选项设置为 True，并为新池或现有池设置 target_size_ratio 值：**

示例

```
openstack_pools:
   - {"name": backups, "target_size_ratio": 0.1, "pg_autoscale_mode": True, "application":
rbd}
   - {"name": volumes, "target_size_ratio": 0.5, "pg_autoscale_mode": True, "application":
rbd}
   - {"name": vms,     "target_size_ratio": 0.2, "pg_autoscale_mode": True, "application": rbd}
   - {"name": images,  "target_size_ratio": 0.2, "pg_autoscale_mode": True, "application": rbd}
```

注意

target_size_ratio 值是相对于存储集群中其他池的权重百分比。

3. **保存对 group_vars/all.yml 文件的更改。**

4. 运行适当的 **Ansible playbook:**

裸机部署

```
[ansible@admin ceph-ansible]$ ansible-playbook site.yml -i hosts
```

容器部署

```
[ansible@admin ceph-ansible]$ ansible-playbook site-container.yml -i hosts
```

5. **Ansible playbook 完成后，从 Ceph 监控节点检查自动扩展状态：**

```
[user@mon ~]$ ceph osd pool autoscale-status
```

5.12. 其它资源

- **Ansible 文档**

第 6 章 容器化 CEPH 守护进程的共存

本节描述：

- colocation 如何工作及其优点
- 如何为 colocated 守护进程设置专用资源

6.1. COLOCATION 如何工作及其优点

您可以在同一个节点上并置容器化 Ceph 守护进程。以下是合并某些 Ceph 服务的优点：

- 规模小，产品总购置成本(TCO)显著改进。
- 对于最低配置，从六个节点减少到三。
- 简化升级.
- 更好的资源隔离。

请参阅知识库文章 *Red Hat Ceph Storage: 支持的配置* 以了解更多有关在 Red Hat Ceph Storage 集群中并置守护进程的信息。

Colocation 工作方式

您可以将以下列表中的一个守护进程与 OSD 守护进程 (ceph-osd) 并置，方法是将同一节点添加到 Ansible 清单文件中的相应部分中。

- Ceph 元数据服务器 (ceph-mds)
- Ceph monitor (ceph-mon) 和 Ceph 管理器 (ceph-mgr) 守护进程

- NFS Ganesha (nfs-ganesha)
- RBD 镜像(rbd-mirror)
- iSCSI 网关(iscsigw)

从红帽 Ceph 存储 4.2 开始，元数据服务器(MDS)可以和一个额外的扩展守护进程在一起。

此外，对于 Ceph 对象网关(comma)或 Grafana，您可以将 OSD 守护进程与上述列表中的守护进程共存，不包括 RBD mirror.z，例如，以下是有效的五个节点共存配置：

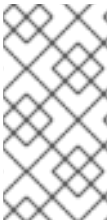
节点	Daemon	Daemon	Daemon
node1	OSD	Monitor	Grafana
node2	OSD	Monitor	RADOS Gateway
node3	OSD	Monitor	RADOS Gateway
node4	OSD	Metadata Server	
node5	OSD	Metadata Server	

如上述设置一样部署一个五个节点集群，请配置 Ansible 清单文件，如下所示：

带有并置守护进程的 Ansible 清单文件

```
[grafana-server]
node1

[mons]
node[1:3]

[mgrs]
node[1:3]

[osds]
node[1:5]
```

```
[rgws]
node[2:3]

[mdss]
node[4:5]
```

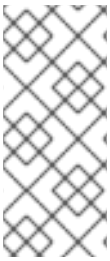


注意

因为 ceph-mon 和 ceph-mgr 可以一起工作，所以不能把两个独立的守护进程计数为两个独立的守护进程。

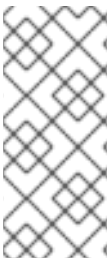
注意

基于 Cockpit 的安装不支持 Colocating Grafana。使用 ceph-ansible 配置存储集群。



注意

红帽建议将 Ceph 对象网关与 OSD 容器共存以提高性能。要在不增加成本的情况下获得最高性能，请在 group_vars/all.yml 中设置 radosgw_num_instances: 2 来使用两个网关。如需更多信息，请参阅 Red Hat Ceph Storage RGW 部署策略和大小调整指导。



注意

需要适当的 CPU 和网络资源才能将 Grafana 与另外两个容器并置在一起。如果发生资源耗尽，只将 Grafana 与 monitor 共存，如果仍然发生资源耗尽，请在专用节点上运行 Grafana。

图 6.1 “colocated Daemons” 和 图 6.2 “非并置守护进程” 镜像显示具有 colocated 和非并置守护进程的集群之间的区别。

图 6.1. colocated Daemons

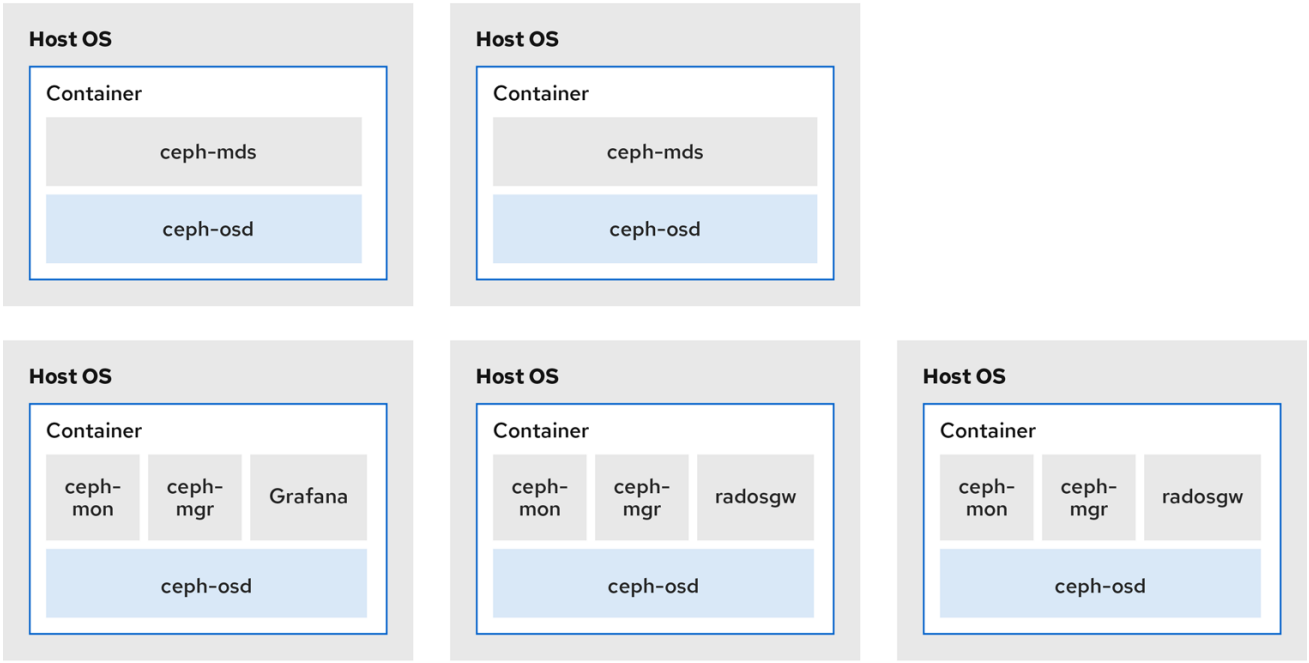


图 6.2. 非并置守护进程

当您在同一节点上并置多个容器化 Ceph 守护进程时，ceph-ansible playbook 会为每个节点保留专用的 CPU 和 RAM 资源。默认情况下，ceph-ansible 使用 *Red Hat Ceph Storage 硬件指南*中推荐的最小硬件章节中列出的值。要了解如何更改默认值，请参阅 Colocated Daemons 的 Setting Dedicated Resources 部分。

6.2. 为 COLOCATED DAEMONS 设置 DEDICATED 资源

当在同一节点上并置两个 Ceph 守护进程时，ceph-ansible playbook 会为每个守护进程保留 CPU 和 RAM 资源。ceph-ansible 使用的默认值在 Red Hat Ceph Storage 硬件选择指南中的推荐的最下硬件章节中列出。若要更改默认值，可在部署 Ceph 守护进程时设置所需的参数。

流程

1. **若要更改守护进程的默认 CPU 限值，可在部署守护进程时设置适当 .yml 配置文件中的 ceph_*daemon-type*_docker_cpu_limit 参数。详情请查看下表。**

Daemon	参数	配置文件
OSD	**ceph_osd_docker_cpu_limit**	**osds.yml**
MDS	**ceph_mds_docker_cpu_limit**	**mdss.yml**
RGW	**ceph_rgw_docker_cpu_limit**	**rgws.yml**

例如，要将 Ceph 对象网关的默认 CPU 限值更改为 2，请按以下方式编辑 /usr/share/ceph-ansible/group_vars/rgws.yml 文件：

```
ceph_rgw_docker_cpu_limit: 2
```

2. **要更改 OSD 守护进程的默认 RAM，请在部署守护进程时设置 /usr/share/ceph-ansible/group_vars/all.yml 文件中的 osd_memory_target。例如，将 OSD RAM 限制为 6 GB：**

```
ceph_conf_overrides:
  osd:
    osd_memory_target=6000000000
```

重要

在超融合基础架构(HCI)配置中，您还可以使用 osds.yml 配置文件中的 ceph_osd_docker_memory_limit 参数来更改 Docker 内存 CGroup 限制。在这种情况下，将 ceph_osd_docker_memory_limit 设置为比 osd_memory_target 高 50%，因此 CGroup 的限制比 HCI 配置的默认值更高。例如，如果 osd_memory_target 设置为 6 GB，则将 ceph_osd_docker_memory_limit 设置为 9 GB：

```
ceph_osd_docker_memory_limit: 9g
```

其它资源

- /usr/share/ceph-ansible/group_vars/ 目录中的配置文件示例

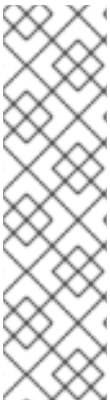
6.3. 其它资源

- *Red Hat Ceph Storage 硬件选择指南*

第 7 章 升级 RED HAT CEPH STORAGE 集群

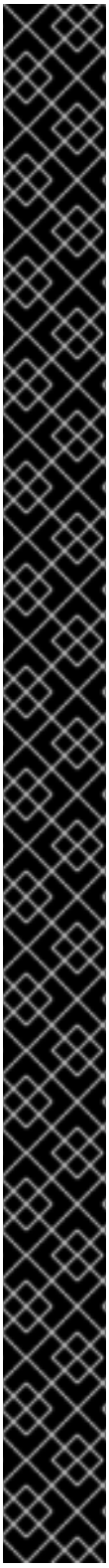
作为存储管理员，您可以将 Red Hat Ceph Storage 集群升级到新的主版本或新的次版本，或者仅对当前版本应用异步更新。rolling_update.yml Ansible playbook 为 Red Hat Ceph Storage 裸机或容器化部署执行升级。Ansible 按照以下顺序升级 Ceph 节点：

- 监控节点
- MGR 节点
- OSD 节点
- MDS 节点
- Ceph 对象网关节点
- 所有其他 Ceph 客户端节点



注意

从 Red Hat Ceph Storage 3.1 开始，添加了新的 Ansible playbook，以便在使用对象网关和基于 NVMe 的 SSD（及 SATA SSD）时优化存储的性能。Playbook 通过将日志和 bucket 索引放在 SSD 上来实现此目的；与将所有日志放在一个设备上相比，这提高了性能。这些 playbook 设计为在安装 Ceph 时使用。现有的 OSD 继续工作，升级期间不需要额外的步骤。无法升级 Ceph 集群，同时重新配置 OSD 以优化存储。若要将不同的设备用于日志或 bucket 索引，需要重新调配 OSD。如需更多信息，请参阅*生产环境指南中的 Ceph 对象网关*中的*最佳使用 NVMe*。



重要

当将 Red Hat Ceph Storage 集群从以前支持的版本升级到 4.2z2 时，升级会在 HEALTH_WARN 状态下完成存储集群，指出 monitor 允许不安全的 global_id 重新声明。这是因为一个补丁的 CVE，其详细信息包括在 *CVE-2021-20288* 中。这个问题由 Red Hat Ceph Storage 4.2z2 的 CVE 解决。

拒绝健康警告的建议：

1. **通过检查 AUTH_INSECURE_GLOBAL_ID_RECLAIM 警报的 ceph health detail 输出，识别尚未更新的客户端。**

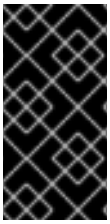
2. **将所有客户端升级到 Red Hat Ceph Storage 4.2z2 版本。**

3. **当验证所有客户端都已更新，并且客户端不再存在 *AUTH_INSECURE_GLOBAL_ID_RECLAIM* 警报后，将 auth_allow_insecure_global_id_reclaim 设置为 false当此选项被设置为 false 时，在网络中断中断后，一个未修补的客户端无法重新连接到存储集群，或者可以在超时时续订其验证票据，默认为 72 小时。**

   语法

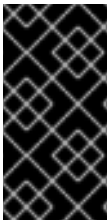
   ```
   ceph config set mon auth_allow_insecure_global_id_reclaim false
   ```

4. **确保没有使用 AUTH_INSECURE_GLOBAL_ID_RECLAIM 警报列出客户端。**



重要

rolling_update.yml playbook 包含 serial 变量，用于调整要同时更新的节点数量。红帽强烈建议使用默认值 (1)，以确保 Ansible 逐一升级集群节点。



重要

如果升级在任何点上失败，请使用 ceph status 命令检查集群状态以了解升级失败的原因。如果您不确定故障原因及如何解决，请联系 *红帽支持*以获得帮助。

警告

如果将多站点设置从 Red Hat Ceph Storage 3 升级到 Red Hat Ceph Storage 4，请遵循以下建议，否则复制可能会中断。在运行 rolling_update.yml 前，在 all.yml 中设置 rgw_multisite: false。升级后不要重新启用 rgw_multisite。只有在升级后需要添加新网关时才使用它。仅将版本 3.3z5 或更高版本的 Red Hat Ceph Storage 3 集群升级到 Red Hat Ceph Storage 4。如果您无法更新到 3.3z5 或更高版本，请在升级集群前禁用站点间同步。若要禁用同步，可设置 rgw_run_sync_thread = false 并重新启动 RADOS 网关守护进程。首先升级主集群。升级到 Red Hat Ceph Storage 4.1 或更高版本。要查看与 3.3z5 相关的软件包版本，请参阅什么是 Red Hat Ceph Storage 版本和对应的 Ceph 软件包版本？有关如何禁用同步的说明，请参阅如何临时禁用 RGW 多站点同步？

警告

当使用 Ceph 对象网关并从 Red Hat Ceph Storage 3.x 升级到 Red Hat Ceph Storage 4.x 时，前端会自动从 CivetWeb 更改为 Beast，这是新的默认值。如需更多信息，请参阅对象网关配置和管理指南中的配置。

警告

如果使用 RADOS 网关，Ansible 会将前端从 CivetWeb 切换到 Beast。在这一过程中，RGW 实例名称从 rgw.*HOSTNAME* 更改为 rgw.*HOSTNAME*.rgw0。由于名称更改，Ansible 不会更新 ceph.conf 中的现有 RGW 配置，而是附加一个默认配置，保留原 CivetWeb 基于 RGW 设置，但它不会被使用。然后，自定义 RGW 配置更改将丢失，这可能会造成 RGW 服务中断。要避免这种情况，请在升级前将现有 RGW 配置添加到 all.yml 的 ceph_conf_overrides 部分，但通过附加 .rgw0 来更改 RGW 实例名称，然后重启 RGW 服务。这将在升级后保留非默认 RGW 配置更改。如需有关 ceph_conf_overrides 的信息，请参阅覆盖 Ceph 默认设置。

7.1. 支持的 RED HAT CEPH STORAGE 升级场景

红帽支持以下升级方案：

阅读*裸机*的表，并*容器化*以了解集群必须处于升级前的状态才能进入升级后状态。

使用 ceph-ansible 执行裸机和容器化升级，其中裸机或主机操作系统不会更改主要版本。ceph-ansible 不支持从 Red Hat Enterprise Linux 7 升级到 Red Hat Enterprise Linux 8。要作为升级 Red Hat Ceph Storage 的一部分将裸机操作系统从 Red Hat Enterprise Linux 7.9 升级到 Red Hat Enterprise Linux 8.4，请参阅 Red Hat Ceph Storage *安装指南*中的*手动升级 Red Hat Ceph Storage 集群和操作系统* 部分。

注意

要将集群升级到 Red Hat Ceph Storage 4，红帽建议您的集群使用最新版本的 Red Hat Ceph Storage 3。要了解最新版本的 Red Hat Ceph Storage，请参见 *什么是 Red Hat Ceph Storage 版本？*如需更多信息，知识库文章。

表 7.1. 支持裸机部署的升级场景

预升级状态		升级后状态	
Red Hat Enterprise Linux 版本	Red Hat Ceph Storage 版本	Red Hat Enterprise Linux 版本	Red Hat Ceph Storage 版本
7.6	3.3	7.9	4.2

预升级状态		升级后状态	
7.6	3.3	8.4	4.2
7.7	3.3	7.9	4.2
7.7	4.0	7.9	4.2
7.8	3.3	7.9	4.2
7.8	3.3	8.4	4.2
7.9	3.3	8.4	4.2
8.1	4.0	8.4	4.2
8.2	4.1	8.4	4.2
8.2	4.1	8.4	4.2
8.3	4.1	8.4	4.2

表 7.2. 支持的容器化部署的升级场景

预升级状态			升级后状态		
主机 Red Hat Enterprise Linux 版本	Container Red Hat Enterprise Linux 版本	Red Hat Ceph Storage 版本	主机 Red Hat Enterprise Linux 版本	Container Red Hat Enterprise Linux 版本	Red Hat Ceph Storage 版本
7.6	7.8	3.3	7.9	8.4	4.2
7.7	7.8	3.3	7.9	8.4	4.2
7.7	8.1	4.0	7.9	8.4	4.2
7.8	7.8	3.3	7.9	8.4	4.2
8.1	8.1	4.0	8.4	8.4	4.2
8.2	8.2	4.1	8.4	8.4	4.2
8.3	8.3	4.1	8.4	8.4	4.2

7.2. 准备升级

在开始升级 Red Hat Ceph Storage 前，需要完成一些任务。这些步骤适用于 Red Hat Ceph Storage 集群的裸机和容器部署，除非为其中一个集群指定。

重要

您只能升级到最新版本的 Red Hat Ceph Storage 4。例如，如果版本 4.1 可用，则无法从 3 升级到 4.0；您必须直接升级到 4.1。

重要

如果使用 FileStore 对象存储，在从 Red Hat Ceph Storage 3 升级到 Red Hat Ceph Storage 4 后，您必须迁移到 BlueStore。

重要

当同时将 Red Hat Enterprise Linux 7 升级到 Red Hat Enterprise Linux 8 时，您无法使用 ceph-ansible 升级 Red Hat Ceph Storage。您必须继续使用 Red Hat Enterprise Linux 7。要升级操作系统，请参阅手动升级 Red Hat Ceph Storage 集群和操作系统。

重要

对于 Red Hat Ceph Storage 4.2z2 及更新的版本，默认情况下 bluefs_buffered_io 选项被设置为 True。这个选项使 BlueFS 能够在某些情况下执行缓冲的读取，并允许内核页面缓存作为辅助缓存进行读取，如 RocksDB 块读取。例如，如果 RocksDB 块缓存不足以在 OMAP 迭代期间保存所有块，则可以从页面缓存而不是磁盘中读取它们。当 osd_memory_target 太小而无法存放块缓存中的所有条目时，这可显著提高性能。当前启用 bluefs_buffered_io 并禁用系统级别交换可防止性能下降。

先决条件

- 对存储集群中所有节点的根级别访问权限。
- 存储集群中所有节点的系统时钟会被同步。如果 monitor 节点没有同步，升级过程可能无法正确完成。
- 如果从版本 3 升级，版本 3 集群会升级到最新版本的 Red Hat Ceph Storage 3。

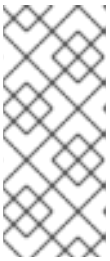
- 在升级到版本 4 之前，如果 Prometheus 节点导出器服务正在运行，请停止该服务：

示例

```
[root@mon ~]# systemctl stop prometheus-node-exporter.service
```

重要

这是一个已知问题，将在即将发布的 Red Hat Ceph Storage 发行版中解决。有关此问题的更多详细信息，请参阅红帽知识库文章。



注意

对于在升级过程中无法访问互联网的 Bare-metal 或 Container Red Hat Ceph Storage 集群节点，请按照 *Red Hat Ceph Storage 安装指南中* 中的 *将 Red Hat Ceph Storage 节点注册到 CDN* 一节中的步骤附加订阅。

流程

1. 以 root 用户身份登录存储集群中的所有节点。

2. 如果 Ceph 节点没有连接到 Red Hat Content Delivery Network (CDN)，您可以使用 ISO 镜像来升级 Red Hat Ceph Storage，方法是使用最新版本的 Red Hat Ceph Storage 新本地存储库。

3. 如果将 Red Hat Ceph Storage 从版本 3 升级到版本 4，请删除现有的 Ceph 控制面板安装。

   a. 在 Ansible 管理节点上，切换到 cephmetrics-ansible 目录：

   ```
   [root@admin ~]# cd /usr/share/cephmetrics-ansible
   ```

b. 运行 **purge.yml playbook** 以删除现有的 **Ceph** 仪表板安装：

```
[root@admin cephmetrics-ansible]# ansible-playbook -v purge.yml
```

4. 如果将 **Red Hat Ceph Storage** 从版本 **3** 升级到版本 **4**，请在 **Ansible** 管理节点上启用 **Ceph** 和 **Ansible** 存储库：

Red Hat Enterprise Linux 7

```
[root@admin ~]# subscription-manager repos --enable=rhel-7-server-rhceph-4-tools-rpms --enable=rhel-7-server-ansible-2.9-rpms
```

Red Hat Enterprise Linux 8

```
[root@admin ~]# subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms --enable=ansible-2.9-for-rhel-8-x86_64-rpms
```

5. 在 **Ansible** 管理节点上，确保安装了最新版本的 **ansible** 和 **ceph-ansible** 软件包。

Red Hat Enterprise Linux 7

```
[root@admin ~]# yum update ansible ceph-ansible
```

Red Hat Enterprise Linux 8

```
[root@admin ~]# dnf update ansible ceph-ansible
```

6. 编辑 **infrastructure-playbooks/rolling_update.yml playbook，并将 health_osd_check_retries 和 health_osd_check_delay** 值分别改为 **50 和 30 ：**

```
health_osd_check_retries: 50
health_osd_check_delay: 30
```

对于每个 **OSD** 节点，这些值可使 **Ansible** 等待最多 **25** 分钟，并且每隔 **30** 秒检查存储集群运行状况，等待继续升级过程。

注意

根据存储集群的已用存储容量，调整 **health_osd_check_retries** 选项的值。例如，如果您在 **436 TB** 中使用 **218 TB**，基本上使用 **50%** 的存储容量，然后将 **health_osd_check_retries** 选项设置为 **50**。

7. 如果要升级的存储集群包含使用 **exclusive-lock** 功能的 **Ceph** 块设备镜像，请确保所有 **Ceph** 块设备用户都有将客户端列入黑名单的权限：

```
ceph auth caps client.ID mon 'allow r, allow command "osd blacklist"' osd
'EXISTING_OSD_USER_CAPS'
```

8. 如果存储集群最初使用 **Cockpit** 安装，请在 **/usr/share/ceph-ansible** 目录中创建一个符号链接到 **Cockpit** 创建它的清单文件，位于 **/usr/share/ansible-runner-service/inventory/hosts ：**

   a. 进入 **/usr/share/ceph-ansible** 目录：

   ```
   # cd /usr/share/ceph-ansible
   ```

   b. 创建符号链接：

   ```
   # ln -s /usr/share/ansible-runner-service/inventory/hosts hosts
   ```

9. 要使用 **ceph-ansible** 升级集群，请在 **etc/ansible/hosts** 目录中创建符号链接到 **hosts** 清单文件：

```
# ln -s /etc/ansible/hosts hosts
```

10. **如果存储集群最初使用 Cockpit 安装，请将 Cockpit 生成的 SSH 密钥复制到 Ansible 用户的 ~/.ssh 目录中：**

    a. **复制密钥：**

    ```
    # cp /usr/share/ansible-runner-service/env/ssh_key.pub
    /home/ANSIBLE_USERNAME/.ssh/id_rsa.pub
    # cp /usr/share/ansible-runner-service/env/ssh_key
    /home/ANSIBLE_USERNAME/.ssh/id_rsa
    ```

    将 *ANSIBLE_USERNAME* 替换为 Ansible 的用户名，通常是 admin。

    示例

    ```
    # cp /usr/share/ansible-runner-service/env/ssh_key.pub /home/admin/.ssh/id_rsa.pub
    # cp /usr/share/ansible-runner-service/env/ssh_key /home/admin/.ssh/id_rsa
    ```

    b. **在密钥文件中设置适当的所有者、组群和权限：**

    ```
    # chown ANSIBLE_USERNAME:_ANSIBLE_USERNAME_
    /home/ANSIBLE_USERNAME/.ssh/id_rsa.pub
    # chown ANSIBLE_USERNAME:_ANSIBLE_USERNAME_
    /home/ANSIBLE_USERNAME/.ssh/id_rsa
    # chmod 644 /home/ANSIBLE_USERNAME/.ssh/id_rsa.pub
    # chmod 600 /home/ANSIBLE_USERNAME/.ssh/id_rsa
    ```

    将 *ANSIBLE_USERNAME* 替换为 Ansible 的用户名，通常是 admin。

    示例

    ```
    # chown admin:admin /home/admin/.ssh/id_rsa.pub
    # chown admin:admin /home/admin/.ssh/id_rsa
    # chmod 644 /home/admin/.ssh/id_rsa.pub
    ```

```
# chmod 600 /home/admin/.ssh/id_rsa
```

其它资源

- 详情请参阅*启用 Red Hat Ceph Storage 存储库*。
- 有关时钟同步和时钟偏移的更多信息，请参阅 ***Red Hat Ceph Storage 故障排除指南中的 Clock Skew*** 部分。

7.3. 使用 ANSIBLE 升级存储集群

使用 Ansible 部署工具，您可以通过执行滚动升级来升级 Red Hat Ceph Storage 集群。除非另有说明，否则这些步骤适用于裸机和容器部署。

先决条件

- 对 Ansible 管理节点的根级别访问权限.
- ansible 用户帐户。

流程

1. 进入 /usr/share/ceph-ansible/ 目录：

   示例

   ```
   [root@admin ~]# cd /usr/share/ceph-ansible/
   ```

2. 如果从 Red Hat Ceph Storage 3 升级到 Red Hat Ceph Storage 4，请备份 group_vars/all.yml、group_vars/osds.yml 和 group_vars/clients.yml 文件：

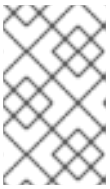
```
[root@admin ceph-ansible]# cp group_vars/all.yml group_vars/all_old.yml
[root@admin ceph-ansible]# cp group_vars/osds.yml group_vars/osds_old.yml
[root@admin ceph-ansible]# cp group_vars/clients.yml group_vars/clients_old.yml
```

3. **如果从 Red Hat Ceph Storage 3 升级到 Red Hat Ceph Storage 4，请分别将 group_vars/all.yml.sample、group_vars/osds.yml.sample 和 group_vars/clients.yml.sample 文件重命名为 group_vars/all.yml、group_vars/osds.yml 和 group_vars/clients.yml。基于之前备份的副本的更改，打开并相应地编辑它们。**

```
[root@admin ceph-ansible]# cp group_vars/all.yml.sample group_vars/all.yml
[root@admin ceph-ansible]# cp group_vars/osds.yml.sample group_vars/osds.yml
[root@admin ceph-ansible]# cp group_vars/clients.yml.sample group_vars/clients.yml
```

4. **编辑 group_vars/osds.yml 文件。添加并设置以下选项：**

```
nb_retry_wait_osd_up: 60
delay_wait_osd_up: 10
```



注意

这些是默认值；您可以根据用例修改值。

5. **如果升级到 Red Hat Ceph Storage 4 的一个新次版本，请验证 group_vars/all.yml 中 grafana_container_image 的值与 group_vars/all.yml.sample 中的值相同。如果它不同，请将其编辑为：**

示例

```
grafana_container_image: registry.redhat.io/rhceph/rhceph-4-dashboard-rhel8:4
```

注意

显示的镜像路径包含在 ceph-ansible 版本 4.0.23-1 中。

6. **从示例文件中复制最新的 site.yml 或 site-container.yml 文件：**

   a. **对于裸机部署：**

   ```
   [root@admin ceph-ansible]# cp site.yml.sample site.yml
   ```

   b. **对于容器部署：**

   ```
   [root@admin ceph-ansible]# cp site-container.yml.sample site-container.yml
   ```

7. **打开 group_vars/all.yml 文件，再编辑下列选项：**

   a. **添加 fetch_directory 选项：**

   ```
   fetch_directory: FULL_DIRECTORY_PATH
   ```

   替换

   - ***FULL_DIRECTORY_PATH*，位置一个可写的位置，如 Ansible 用户的主目录。**

   b. **如果要升级的集群包含任何 Ceph 对象网关节点，请添加 radosgw_interface 选项：**

   ```
   radosgw_interface: INTERFACE
   ```

   替换

   - ***INTERFACE*，具有 Ceph 对象网关节点侦听的接口。**

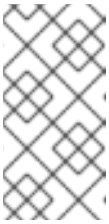
   c. **如果您的当前设置配置了 SSL 证书，您需要编辑以下内容：**

   ```
   radosgw_frontend_ssl_certificate: /etc/pki/ca-trust/extracted/CERTIFICATE_NAME
   radosgw_frontend_port: 443
   ```

   d.

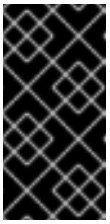
默认 OSD 对象存储为 BlueStore。要保留传统的 OSD 对象存储，您必须将 osd_objectstore 选项明确设置为 filestore ：

```
osd_objectstore: filestore
```



注意

将 osd_objectstore 选项设置为 filestore 时，替换 OSD 将使用 FileStore，而不是 BlueStore。



重要

从 Red Hat Ceph Storage 4 开始，FileStore 是一项弃用的功能。红帽建议将 FileStore OSD 迁移到 BlueStore OSD。

e. 从 Red Hat Ceph Storage 4.1 开始，您必须在 /usr/share/ceph-ansible/group_vars/all.yml 中取消注释或设置 dashboard_admin_password 和 grafana_admin_password。为每个用户设置安全密码.另外，为 dashboard_admin_user 和 grafana_admin_user 设置自定义用户名。

f. 对于裸机和容器部署：

i. 取消注释 upgrade_ceph_packages 选项并将其设置为 True ：

```
upgrade_ceph_packages: True
```

ii. 将 ceph_rhcs_version 选项设置为 4 ：

```
ceph_rhcs_version: 4
```

注意

将 ceph_rhcs_version 选项设置为 4 将拉取最新版本的 Red Hat Ceph Storage 4。

iii. 将 ceph_docker_registry 信息添加到 all.yml ：

语法

```
ceph_docker_registry: registry.redhat.io
ceph_docker_registry_username: SERVICE_ACCOUNT_USER_NAME
ceph_docker_registry_password: TOKEN
```

注意

如果您没有 Red Hat Registry Service Account，请使用 *Registry Service Account 网页* 创建一个。如需了解更多详细信息，请参阅 *Red Hat Container Registry 身份验证* 知识库文章。

注意

除了将服务帐户用于 ceph_docker_registry_username 和 ceph_docker_registry_password 参数外，您还可以使用客户门户凭据，但若要确保安全性，可以对 ceph_docker_registry_password 参数进行加密。如需更多信息，请参阅使用 ansible-vault 加密 Ansible 密码变量。

g. 对于容器部署：

i. 更改 ceph_docker_image 选项以指向 Ceph 4 容器版本：

```
ceph_docker_image: rhceph/rhceph-4-rhel8
```

ii. 更改 ceph_docker_image_tag 选项，使其指向 rhceph/rhceph-4-rhel8 的最新版本：

```
ceph_docker_image_tag: latest
```

8. 如果从 Red Hat Ceph Storage 3 升级到 Red Hat Ceph Storage 4，请打开 Ansible 清单文件进行编辑，默认为 /etc/ansible/hosts，并在 [grafana-server] 部分下添加 Ceph 仪表板节点名称或 IP 地址。如果此部分不存在，还要将本节与节点名称或 IP 地址一起添加。

9. **切换到或以 Ansible 用户身份登录，然后运行 rolling_update.yml playbook:**

```
[ansible@admin ceph-ansible]$ ansible-playbook infrastructure-playbooks/rolling_update.yml
-i hosts
```

重要

不支持将 --limit Ansible 选项与 rolling_update.yml playbook 搭配使用。

10. **作为 RBD 镜像守护进程节点上的 root 用户，手动升级 rbd-mirror 软件包：**

```
[root@rbd ~]# yum upgrade rbd-mirror
```

11. **重启 rbd-mirror 守护进程：**

```
systemctl restart ceph-rbd-mirror@CLIENT_ID
```

12. **验证存储集群的运行状况。**

    a. **对于裸机部署，以 root 用户身份登录监控节点，运行 Ceph status 命令：**

    ```
    [root@mon ~]# ceph -s
    ```

    b. **对于容器部署，请以 root 用户身份登录 Ceph 监控节点。**

        i. **列出所有正在运行的容器：**

Red Hat Enterprise Linux 7

```
[root@mon ~]# docker ps
```

Red Hat Enterprise Linux 8

```
[root@mon ~]# podman ps
```

ii. 检查**健康状**态：

Red Hat Enterprise Linux 7

```
[root@mon ~]# docker exec ceph-mon-MONITOR_NAME ceph -s
```

Red Hat Enterprise Linux 8

```
[root@mon ~]# podman exec ceph-mon-MONITOR_NAME ceph -s
```

替换

- ***MONITOR_NAME*，使用带有上一步中找到的 Ceph monitor 容器的名称。**

  示例

  ```
  [root@mon ~]# podman exec ceph-mon-mon01 ceph -s
  ```

13. **可选：如果从 Red Hat Ceph Storage 3.x 升级到 Red Hat Ceph Storage 4.x，您可能会看到这个健康状况警告：*Legacy BlueStore stats reporting detected on 336 OSD(s)*.这是因为较新的代码计算池统计不同。您可以通过设置 bluestore_fsck_quick_fix_on_mount 参数来解决这个问题。**

    a. **将 bluestore_fsck_quick_fix_on_mount 设置为 true ：**

    示例

    ```
    [root@mon ~]# ceph config set osd bluestore_fsck_quick_fix_on_mount true
    ```

    b. **设置 noout 和 norebalance 标志，以防止 OSD 停机时出现数据移动：**

    示例

    ```
    [root@mon ~]# ceph osd set noout
    [root@mon ~]# ceph osd set norebalance
    ```

    c. **对于 裸机部署，请在存储集群的每个 OSD 节点上重启 ceph-osd.target ：**

    示例

    ```
    [root@osd ~]# systemctl restart ceph-osd.target
    ```

    d. **对于 容器化部署，请在另一个 OSD 后重启各个 OSD，并等待所有放置组都处于 active+clean 状态。**

语法

```
systemctl restart ceph-osd@OSD_ID.service
```

示例

```
[root@osd ~]# systemctl restart ceph-osd@0.service
```

e. **当所有 OSD 被修复时，取消设置 nout 和 norebalance 标记：**

示例

```
[root@mon ~]# ceph osd unset noout
[root@mon ~]# ceph osd unset norebalance
```

f. **当所有 OSD 修复后，将 bluestore_fsck_quick_fix_on_mount 设置为 false ：**

示例

```
[root@mon ~]# ceph config set osd bluestore_fsck_quick_fix_on_mount false
```

g.

可选：裸机部署 的一个替代方法是停止 OSD 服务，使用 ceph-bluestore-tool 命令在 OSD 上运行修复功能，然后启动 OSD 服务：

i. **停止 OSD 服务：**

```
[root@osd ~]# systemctl stop ceph-osd.target
```

ii. **在 OSD 上运行修复功能，指定其实际 OSD ID：**

语法

```
ceph-bluestore-tool --path /var/lib/ceph/osd/ceph-OSDID repair
```

示例

```
[root@osd ~]# ceph-bluestore-tool --path /var/lib/ceph/osd/ceph-2 repair
```

iii. **启动 OSD 服务：**

```
[root@osd ~]# systemctl start ceph-osd.target
```

14. **升级完成后，您可以通过运行 Ansible playbook 将 FileStore OSD 迁移到 BlueStore OSD:**

语法

```
ansible-playbook infrastructure-playbooks/filestore-to-bluestore.yml --limit
OSD_NODE_TO_MIGRATE
```

示例

```
[ansible@admin ceph-ansible]$ ansible-playbook infrastructure-playbooks/filestore-to-bluestore.yml --limit osd01
```

迁移完成后，请执行以下子步骤：

a. **打开以编辑 group_vars/osds.yml 文件，并将 osd_objectstore 选项设置为 bluestore，例如：**

```
osd_objectstore: bluestore
```

b. **如果您使用 lvm_volumes 变量，分别将 journal 和 journal_vg 选项改为 db 和 db_vg，例如：**

之前

```
lvm_volumes:
  - data: /dev/sdb
    journal: /dev/sdc1
  - data: /dev/sdd
    journal: journal1
    journal_vg: journals
```

转换为 Bluestore 后

```
lvm_volumes:
  - data: /dev/sdb
    db: /dev/sdc1
```

```
- data: /dev/sdd
  db: journal1
  db_vg: journals
```

15. **如果在 OpenStack 环境中工作，请更新所有 cephx 用户，以将 RBD 配置文件用于池。以下命令必须以 root 用户身份运行：**

a. **Glance 用户：**

语法

```
ceph auth caps client.glance mon 'profile rbd' osd 'profile rbd
pool=GLANCE_POOL_NAME'
```

示例

```
[root@mon ~]# ceph auth caps client.glance mon 'profile rbd' osd 'profile rbd
pool=images'
```

b. **Cinder 用户：**

语法

```
ceph auth caps client.cinder mon 'profile rbd' osd 'profile rbd
pool=CINDER_VOLUME_POOL_NAME, profile rbd pool=NOVA_POOL_NAME, profile
rbd-read-only pool=GLANCE_POOL_NAME'
```

示例

```
[root@mon ~]# ceph auth caps client.cinder mon 'profile rbd' osd 'profile rbd
pool=volumes, profile rbd pool=vms, profile rbd-read-only pool=images'
```

c. **OpenStack 常规用户：**

语法

```
ceph auth caps client.openstack mon 'profile rbd' osd 'profile rbd-read-only
pool=CINDER_VOLUME_POOL_NAME, profile rbd pool=NOVA_POOL_NAME, profile
rbd-read-only pool=GLANCE_POOL_NAME'
```

示例

```
[root@mon ~]# ceph auth caps client.openstack mon 'profile rbd' osd 'profile rbd-read-
only pool=volumes, profile rbd pool=vms, profile rbd-read-only pool=images'
```

重要

在执行任何实时客户端迁移前，进行这些 CAPS 更新。这使得客户端能够使用内存中运行的新库，从而导致旧 CAPS 设置从缓存中丢弃并应用新的 RBD 配置集设置。

16. 可选：在客户端节点上，重新启动依赖于 Ceph 客户端侧库的任何应用。

注意

如果您要升级运行 QEMU 或 KVM 实例的 OpenStack Nova 计算节点，或使用专用 QEMU 或 KVM 客户端，请停止并启动 QEMU 或 KVM 实例，因为在此情况下重启实例不起作用。

其它资源

- 如需了解更多详细信息，请参阅 *了解 limit 选项*。
- 如需更多信息，请参阅 *Red Hat Ceph Storage Administration Guide* 中的 *How to migrate the object store from FileStore to BlueStore*。
- 如需了解更多详细信息，请参阅 *ceph-upgrade cluster status 报告 'Legacy BlueStore stats reporting'*。

7.4. 使用命令行界面升级存储集群

您可以在存储集群运行时从 Red Hat Ceph Storage 3.3 升级到 Red Hat Ceph Storage 4。这些版本之间的重要区别在于，Red Hat Ceph Storage 4 默认使用 msgr2 协议，该协议使用端口 3300。如果没有打开，集群将发出 HEALTH_WARN 错误。

升级存储集群时需要考虑以下限制：

- 默认情况下，Red Hat Ceph Storage 4 使用 msgr2 协议。确保 Ceph 监控节点上打开了端口 3300
- 将 ceph-monitor 守护进程从 Red Hat Ceph Storage 3 升级到 Red Hat Ceph Storage 4 后，Red Hat Ceph Storage 3 ceph-osd 守护进程无法创建新 OSD，直到您将它们升级到 Red Hat Ceph Storage 4。
- 不要在升级进行时创建任何池。

先决条件

先决条件

- **Ceph 监控器、OSD 和对象网关节点的根级别访问权限。**

流程

1. **在运行 Red Hat Ceph Storage 3 时，确保集群至少完成了所有 PG 的完全清理。如果不这样做，会导致 monitor 守护进程在启动时拒绝加入仲裁，从而使它们无法运行。要确保集群至少完成所有 PG 的一个完整清理，请执行以下操作：**

```
# ceph osd dump | grep ^flags
```

要从 Red Hat Ceph Storage 3 升级到 Red Hat Ceph Storage 4，OSD map 必须包含 restore_deletes 和 purged_snapdirs 标志。

2. **确保集群处于健康而干净的状态。**

```
ceph health
HEALTH_OK
```

3. **对于运行 ceph-mon 和 ceph-manager 的节点，请执行：**

```
# subscription-manager repos --enable=rhel-7-server-rhceph-4-mon-rpms
```

启用 Red Hat Ceph Storage 4 软件包后，在每个 ceph-mon 和 ceph-manager 节点上执行下列操作：

```
# firewall-cmd --add-port=3300/tcp
# firewall-cmd --add-port=3300/tcp --permanent
# yum update -y
# systemctl restart ceph-mon@<mon-hostname>
# systemctl restart ceph-mgr@<mgr-hostname>
```

将 <mon-hostname> 和 <mgr-hostname> 替换为目标主机的主机名。

4. **在升级 OSD 之前，请在 Ceph 监控节点上设置 noout 和 nodeep-scrub 标志，以防止在升级过程中重新平衡 OSD。**

```
# ceph osd set noout
# ceph osd det nodeep-scrub
```

5. **在每个 OSD 节点上执行：**

```
# subscription-manager repos --enable=rhel-7-server-rhceph-4-osd-rpms
```

启用 Red Hat Ceph Storage 4 软件包后，更新 OSD 节点：

```
# yum update -y
```

对于节点上运行的每个 OSD 守护进程，执行：

```
# systemctl restart ceph-osd@<osd-num>
```

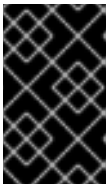
将 <osd-num> 替换为要重启的 osd 号。在继续下一 OSD 节点之前，确保节点上的所有 OSD 都已重启。

6. **如果存储集群中有任何 OSD 使用 ceph-disk 部署，指示 ceph-volume 启动守护进程。**

```
# ceph-volume simple scan
# ceph-volume simple activate --all
```

7. **仅启用 Nautilus 功能：**

```
# ceph osd require-osd-release nautilus
```



重要

如果无法执行此步骤，OSD 将无法在启用 msgr2 后进行通信。

8. **升级所有 OSD 节点后，取消设置 Ceph 监控节点上的 noout 和 nodeep-scrub 标志。**

```
# ceph osd unset noout
# ceph osd unset nodeep-scrub
```

9. **将任何现有的 CRUSH bucket 切换到最新的 bucket 类型 straw2。**

```
# ceph osd getcrushmap -o backup-crushmap
# ceph osd crush set-all-straw-buckets-to-straw2
```

10. **从 Red Hat Ceph Storage 3 升级到 Red Hat Ceph Storage 4 后，即可执行以下步骤：**

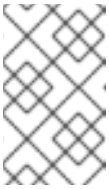
    a. **启用消息传递 v2 协议 msgr2：**

    ```
    ceph mon enable-msgr2
    ```

    这将指示绑定到 6789 的旧默认端口的所有 Ceph 监控器也绑定到 3300 的新端口。

    b. **验证 monitor 的状态：**

    ```
    ceph mon dump
    ```

    注意

    运行 nautilus OSD 不会自动绑定到其 v2 地址。必须重启它们。

11. **对于从 Red Hat Ceph Storage 3 升级到 Red Hat Ceph Storage 4 的每个主机，将 ceph.conf 文件更新为没有指定任何监控端口，或引用 v2 和 v1 地址和端口。**

12. **将 ceph.conf 文件中的任何配置选项导入到存储集群的配置数据库中。**

示例

```
[root@mon ~]# ceph config assimilate-conf -i /etc/ceph/ceph.conf
```

a.

检查存储集群的配置数据库。

示例

```
[root@mon ~]# ceph config dump
```

b.

可选：升级到 Red Hat Ceph Storage 4 后，为每个主机创建一个最小的 ceph.conf 文件：

示例

```
[root@mon ~]# ceph config generate-minimal-conf > /etc/ceph/ceph.conf.new
[root@mon ~]# mv /etc/ceph/ceph.conf.new /etc/ceph/ceph.conf
```

13.

在 Ceph 对象网关节点上执行：

```
# subscription-manager repos --enable=rhel-7-server-rhceph-4-tools-rpms
```

启用 Red Hat Ceph Storage 4 软件包后，更新节点并重启 ceph-rgw 守护进程：

```
# yum update -y
# systemctl restart ceph-rgw@<rgw-target>
```

将 <rgw-target> 替换为要重启的 rgw 目标。

14.

对于管理节点，执行：

```
# subscription-manager repos --enable=rhel-7-server-rhceph-4-tools-rpms
# yum update -y
```

15. 确保集群处于健康而干净的状态。

```
# ceph health
HEALTH_OK
```

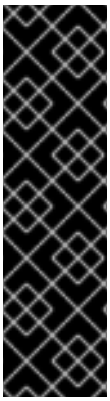
16. 可选：在客户端节点上，重新启动依赖于 Ceph 客户端侧库的任何应用。

注意

如果您要升级运行 QEMU 或 KVM 实例的 OpenStack Nova 计算节点，或使用专用 QEMU 或 KVM 客户端，请停止并启动 QEMU 或 KVM 实例，因为在此情况下重启实例不起作用。

7.5. 手动升级 CEPH 文件系统元数据服务器节点

您可以在运行 Red Hat Enterprise Linux 7 或 8 的 Red Hat Ceph Storage 集群中手动升级 Ceph 文件系统 (CephFS) 元数据服务器 (MDS) 软件。



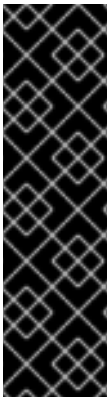
重要

在升级存储集群前，请将活跃 MDS 的数量减少为每个文件系统一个。这消除了多个 MDS 之间可能存在的版本冲突。另外，在升级前关闭所有待机节点。

这是因为 MDS 集群没有内置的版本或文件系统标志。如果没有这些功能，多个 MDS 可能会使用不同版本的 MDS 软件进行通信，并可能导致断言或其他故障发生。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 节点使用 Red Hat Ceph Storage 版本 3.3z64 或 4.1。
- 对存储集群中所有节点的根级别访问权限。



重要

底层 XFS 文件系统必须格式化为支持 ftype=1 或 d_type。运行 xfs_info /var 命令以确保 ftype 设置为 1。如果 ftype 的值不是 1，请附加新磁盘或创建卷。在此新设备之上，创建新的 XFS 文件系统并将其挂载到 /var/lib/containers。

从 Red Hat Enterprise Linux 8.0 开始，mkfs.xfs 默认启用 ftype=1。

流程

1. **将活跃 MDS 的数量减少到 1：**

语法

```
ceph fs set FILE_SYSTEM_NAME max_mds 1
```

示例

```
[root@mds ~]# ceph fs set fs1 max_mds 1
```

2. **等待集群停止所有 MDS 等级。当所有 MDS 停止后，仅排名 0 才处于活动状态。剩余的操作应处于待机模式。检查文件系统的状态：**

```
[root@mds ~]# ceph status
```

3. **使用 systemctl 关闭所有备用 MDS：**

```
[root@mds ~]# systemctl stop ceph-mds.target
```

4. **确认只有一个 MDS 在线，并且已为您的文件系统排名 0：**

```
[root@mds ~]# ceph status
```

5. **如果您要从 RHEL 7 上的 Red Hat Ceph Storage 3 升级，请禁用 Red Hat Ceph Storage 3 工具存储库并启用 Red Hat Ceph Storage 4 工具存储库：**

```
[root@mds ~]# subscription-manager repos --disable=rhel-7-server-rhceph-3-tools-rpms
[root@mds ~]# subscription-manager repos --enable=rhel-7-server-rhceph-4-tools-rpms
```

6. **更新节点并重启 ceph-mds 守护进程：**

```
[root@mds ~]# yum update -y
[root@mds ~]# systemctl restart ceph-mds.target
```

7. **为待机守护进程跟踪相同的进程。禁用并启用工具存储库，然后升级并重启每个待机 MDS:**

```
[root@mds ~]# subscription-manager repos --disable=rhel-7-server-rhceph-3-tools-rpms
[root@mds ~]# subscription-manager repos --enable=rhel-7-server-rhceph-4-tools-rpms
[root@mds ~]# yum update -y
[root@mds ~]# systemctl restart ceph-mds.target
```

8. **当您完成重启所有待机 MDS 后，为存储集群恢复之前 max_mds 的值：**

语法

```
ceph fs set FILE_SYSTEM_NAME max_mds ORIGINAL_VALUE
```

示例

```
[root@mds ~]# ceph fs set fs1 max_mds 5
```

7.6. 其它资源

- 要查看与 3.3z5 相关的软件包版本，请参阅什么是 Red Hat Ceph Storage 版本和对应的 Ceph 软件包版本？

第 8 章 手动升级 RED HAT CEPH STORAGE 集群和操作系统

通常，使用 ceph-ansible 时，无法同时将 Red Hat Ceph Storage 和 Red Hat Enterprise Linux 升级到一个新的主版本。例如，如果您在使用 ceph-ansible，使用 Red Hat Enterprise Linux 7，则必须保留该版本。作为系统管理员，您可以手动执行此操作。

使用本章的内容，把在 Red Hat Enterprise Linux 7.9 上运行的版本 4.1 或 3.3z6 的 Red Hat Ceph Storage 集群手动升级到在 Red Hat Enterprise Linux 8.4 上运行的版本 4.2 Red Hat Ceph Storage 集群。

重要

要将版本为 3.x 或 4.x 的容器化 Red Hat Ceph Storage 集群升级到版本 4.2，请参阅 *Red Hat Ceph Storage 安装指南*的以下三个部分：*支持的 Red Hat Ceph Storage 升级场景*，*准备升级*，以及 *Ansible 升级存储集群*。

要迁移现有的 systemd 模板，请运行 docker-to-podman playbook:

```
[user@admin ceph-ansible]$ ansible-playbook infrastructure-playbooks/docker-to-podman.yml -i hosts
```

其中 user 是 Ansible 用户。

重要

如果节点与多个守护进程并置，请遵循本章中的特定部分，以了解节点中并置的守护进程。例如，与 Ceph 监控守护进程和 OSD 守护进程共存的节点：

请参阅 *手动升级 Ceph 监控节点及其操作系统*，以及 *手动升级 Ceph OSD 节点及其操作系统*。

重要

手动升级 Ceph OSD 节点及其操作系统不适用于加密的 OSD 分区，因为 Leapp 升级实用程序不支持使用 OSD 加密升级。

8.1. 先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 节点正在运行 Red Hat Enterprise Linux 7 7.9。
- 节点使用 Red Hat Ceph Storage 版本 3.3z6 或 4.1
- 访问 Red Hat Enterprise Linux 8.3 的安装源。

8.2. 手动升级 CEPH 监控节点及其操作系统

作为系统管理员，您可以手动将 Red Hat Ceph Storage 集群节点上的 Ceph 监控软件和 Red Hat Enterprise Linux 操作系统同时升级到新的主版本。

重要

一次仅在一个 monitor 节点上执行该步骤。要防止集群访问问题，请确保当前升级的 monitor 节点在继续下一节点*之前*重新正常运作。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 节点正在运行 Red Hat Enterprise Linux 7 7.9。
- 节点使用 Red Hat Ceph Storage 版本 3.3z6 或 4.1
- 访问 Red Hat Enterprise Linux 8.3 的安装源。

流程

1. 停止 monitor 服务：

语法

```
systemctl stop ceph-mon@MONITOR_ID
```

将 ***MONITOR_ID*** 替换为 **monitor 的 ID 号。**

2. **如果使用 Red Hat Ceph Storage 3，请禁用 Red Hat Ceph Storage 3 存储库。**

   a. **禁用工具存储库：**

   ```
   [root@mon ~]# subscription-manager repos --disable=rhel-7-server-rhceph-3-tools-rpms
   ```

   b. **禁用 mon 存储库：**

   ```
   [root@mon ~]# subscription-manager repos --disable=rhel-7-server-rhceph-3-mon-rpms
   ```

3. **如果使用 Red Hat Ceph Storage 4，请禁用 Red Hat Ceph Storage 4 存储库。**

   a. **禁用工具存储库：**

   ```
   [root@mon ~]# subscription-manager repos --disable=rhel-7-server-rhceph-4-tools-rpms
   ```

   b. **禁用 mon 存储库：**

   ```
   [root@mon ~]# subscription-manager repos --disable=rhel-7-server-rhceph-4-mon-rpms
   ```

4. **安装 leapp 实用程序。请参阅从 Red Hat Enterprise Linux 7 升级到 Red Hat Enterprise Linux 8。**

5. **通过 leapp preupgrade 检查运行。请参阅从命令行评估可升级性。**

6. **在 /etc/ssh/sshd_config 中设置 PermitRootLogin yes。**

7. **重启 OpenSSH SSH 守护进程：**

```
[root@mon ~]# systemctl restart sshd.service
```

8. **从 Linux 内核中删除 iSCSI 模块：**

```
[root@mon ~]# modprobe -r iscsi
```

9. **执行 从 RHEL 7 升级到 RHEL 8 的内容，以执行升级。**

10. **重新引导节点。**

11. **为 Red Hat Enterprise Linux 8 启用 Red Hat Ceph Storage 4 的软件仓库。**

    a. **启用工具存储库：**

    ```
    [root@mon ~]# subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
    ```

    b. **启用 mon 存储库：**

    ```
    [root@mon ~]# subscription-manager repos --enable=rhceph-4-mon-for-rhel-8-x86_64-rpms
    ```

12. **安装 ceph-mon 软件包：**

```
[root@mon ~]# dnf install ceph-mon
```

13. **如果管理器服务与 monitor 服务在一起，请安装 ceph-mgr 软件包：**

```
[root@mon ~]# dnf install ceph-mgr
```

14. **从尚未升级的 monitor 节点或已经恢复这些文件的节点恢复 ceph-client-admin.keyring 和 ceph.conf 文件。**

15. **将任何现有的 CRUSH bucket 切换到最新的 bucket 类型 straw2。**

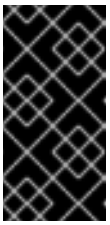
```
# ceph osd getcrushmap -o backup-crushmap
# ceph osd crush set-all-straw-buckets-to-straw2
```

16. **从 Red Hat Ceph Storage 3 升级到 Red Hat Ceph Storage 4 后，即可执行以下步骤：**

    a. **启用消息传递 v2 协议 msgr2：**

    ```
    ceph mon enable-msgr2
    ```

    这将指示绑定到 6789 的旧默认端口的所有 Ceph 监控器也绑定到 3300 的新端口。

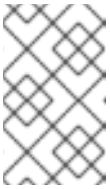
    

    重要

    在执行任何进一步的 Ceph Monitor 配置之前，确保所有 Ceph Monitor 都已从 Red Hat Ceph Storage 3 升级到 Red Hat Ceph Storage 4。

    b. **验证 monitor 的状态：**

    ```
    ceph mon dump
    ```

    注意

    运行 nautilus OSD 不会自动绑定到其 v2 地址。必须重启它们。

17. **对于从 Red Hat Ceph Storage 3 升级到 Red Hat Ceph Storage 4 的每个主机，将 ceph.conf 文件更新为没有指定任何监控端口，或引用 v2 和 v1 地址和端口。将 ceph.conf 文件中的任何配置选项导入到存储集群的配置数据库中。**

示例

```
[root@mon ~]# ceph config assimilate-conf -i /etc/ceph/ceph.conf
```

a. **检查存储集群的配置数据库。**

示例

```
[root@mon ~]# ceph config dump
```

b. **可选：升级到 Red Hat Ceph Storage 4 后，为每个主机创建一个最小的 ceph.conf 文件：**

示例

```
[root@mon ~]# ceph config generate-minimal-conf > /etc/ceph/ceph.conf.new
[root@mon ~]# mv /etc/ceph/ceph.conf.new /etc/ceph/ceph.conf
```

18. **安装 leveldb 软件包：**

```
[root@mon ~]# dnf install leveldb
```

19. **启动监控器服务：**

```
[root@mon ~]# systemctl start ceph-mon.target
```

20. **如果管理器服务与监控服务在一起，也启动管理器服务：**

```
[root@mon ~]# systemctl start ceph-mgr.target
```

21. **验证 monitor 服务是否已恢复，且是否在仲裁数中。**

```
[root@mon ~]# ceph -s
```

在 *services* 下的 *mon:* 行中， 确保该节点列为*已仲裁*，而不是作为*没有仲裁*列出。

示例

```
mon: 3 daemons, quorum ceph4-mon,ceph4-mon2,ceph4-mon3 (age 2h)
```

22. **如果 manager 服务与 monitor 服务在一起，请验证是否也启动：**

```
[root@mon ~]# ceph -s
```

在 *services* 下的 *mgr:* 行中查找管理器的节点名称。

示例

```
mgr: ceph4-mon(active, since 2h), standbys: ceph4-mon3, ceph4-mon2
```

23. **在所有 monitor 节点上重复上述步骤，直到它们都已升级。**

其它资源

- 如需更多信息，请参阅安装指南中的手动升级 Red Hat Ceph Storage 集群和操作系统。

- 如需更多信息，请参阅 从 Red Hat Enterprise Linux 7 升级到 Red Hat Enterprise Linux 8。

8.3. 手动升级 CEPH OSD 节点及其操作系统

作为系统管理员，您可以手动将 Red Hat Ceph Storage 集群节点上的 Ceph OSD 软件和 Red Hat Enterprise Linux 操作系统同时升级到新的主版本。

重要

应当对 Ceph 集群中的每一 OSD 节点执行此步骤，但通常一次仅针对一个 OSD 节点执行此步骤。可以并行执行最多一个值得 OSD 节点的故障域。例如，如果正在使用每个机架复制，可以并行升级整个机架的 OSD 节点。为防止数据访问问题，请确保当前 OSD 节点的 OSD 已恢复正常运作，并且集群的所有 PG 在继续下一 OSD 之前处于 active+clean 状态。

重要

此流程不适用于加密的 OSD 分区，因为 Leapp 升级工具不支持使用 OSD 加密升级。

重要

如果 OSD 是使用 ceph-disk 创建的，并且仍然由 ceph-disk 管理，则必须使用 ceph-volume 接管它们的管理。下面的一个可选步骤将对此进行阐述。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。

- 节点正在运行 Red Hat Enterprise Linux 7 7.9。

- 节点使用 Red Hat Ceph Storage 版本 3.3z6 或 4.0

- 访问 **Red Hat Enterprise Linux 8.3 的安装源。**

流程

1. 设置 **OSD noout 标志，以防止 OSD 在迁移期间被标记为 down：**

```
ceph osd set noout
```

2. 设置 **OSD nobackfill、norecover、norrebalance、noscrub 和 nodeep-scrub 标志，以避免集群出现不必要的负载，并在节点停机时避免任何数据被重新创建：**

```
ceph osd set nobackfill
ceph osd set norecover
ceph osd set norebalance
ceph osd set noscrub
ceph osd set nodeep-scrub
```

3. **正常关闭节点上的所有 OSD 进程：**

```
[root@mon ~]# systemctl stop ceph-osd.target
```

4. **如果使用 Red Hat Ceph Storage 3，请禁用 Red Hat Ceph Storage 3 存储库。**

   a. **禁用工具存储库：**

   ```
   [root@mon ~]# subscription-manager repos --disable=rhel-7-server-rhceph-3-tools-rpms
   ```

   b. **禁用 osd 存储库：**

   ```
   [root@mon ~]# subscription-manager repos --disable=rhel-7-server-rhceph-3-osd-rpms
   ```

5. **如果使用 Red Hat Ceph Storage 4，请禁用 Red Hat Ceph Storage 4 存储库。**

   a. **禁用工具存储库：**

   ```
   [root@mon ~]# subscription-manager repos --disable=rhel-7-server-rhceph-4-tools-rpms
   ```

b. **禁用 osd 存储库：**

```
[root@mon ~]# subscription-manager repos --disable=rhel-7-server-rhceph-4-osd-rpms
```

6. **安装 leapp 实用程序。请参阅从 Red Hat Enterprise Linux 7 升级到 Red Hat Enterprise Linux 8。**

7. **通过 leapp preupgrade 检查运行。请参阅从命令行评估可升级性。**

8. **在 /etc/ssh/sshd_config 中设置 PermitRootLogin yes。**

9. **重启 OpenSSH SSH 守护进程：**

```
[root@mon ~]# systemctl restart sshd.service
```

10. **从 Linux 内核中删除 iSCSI 模块：**

```
[root@mon ~]# modprobe -r iscsi
```

11. **执行 从 RHEL 7 升级到 RHEL 8 的内容，以执行升级。**

12. **重新引导节点。**

13. **为 Red Hat Enterprise Linux 8 启用 Red Hat Ceph Storage 4 的软件仓库。**

a. **启用工具存储库：**

```
[root@mon ~]# subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
```

b. **启用 osd 存储库：**

```
[root@mon ~]# subscription-manager repos --enable=rhceph-4-osd-for-rhel-8-x86_64-
rpms
```

14. **安装 ceph-osd 软件包：**

```
[root@mon ~]# dnf install ceph-osd
```

15. **安装 leveldb 软件包：**

```
[root@mon ~]# dnf install leveldb
```

16. **从尚未升级的节点或已经恢复这些文件的节点恢复 ceph.conf 文件。**

17. **取消设置 noout、nobackfill、norecover、norebalance、noscrub 和 nodeep-scrub 标志：**

```
# ceph osd unset noout
# ceph osd unset nobackfill
# ceph osd unset norecover
# ceph osd unset norebalance
# ceph osd unset noscrub
# ceph osd unset nodeep-scrub
```

18. **将任何现有的 CRUSH bucket 切换到最新的 bucket 类型 straw2。**

```
# ceph osd getcrushmap -o backup-crushmap
# ceph osd crush set-all-straw-buckets-to-straw2
```

19. **可选：如果 OSD 是使用 ceph-disk 创建的，并且仍然由 ceph-disk 管理，则必须使用 ceph-volume 接管它们的管理。**

    a. **挂载每个对象存储设备：**

    语法

```
/dev/DRIVE /var/lib/ceph/osd/ceph-OSD_ID
```

使用存储设备名称和分区号替换 ***DRIVE***。

将 ***OSD_ID*** 替换为 **OSD ID**。

示例

```
[root@mon ~]# mount /dev/sdb1 /var/lib/ceph/osd/ceph-0
```

验证 ***ID_NUMBER*** 是否正确。

语法

```
cat /var/lib/ceph/osd/ceph-OSD_ID/whoami
```

将 ***OSD_ID*** 替换为 **OSD ID**。

示例

```
[root@mon ~]# cat /var/lib/ceph/osd/ceph-0/whoami
0
```

对任何其他对象存储设备重复上述步骤。

b. 扫描新挂载的设备：

语法

```
ceph-volume simple scan /var/lib/ceph/osd/ceph-OSD_ID
```

将 *OSD_ID* 替换为 OSD ID。

示例

```
[root@mon ~]# ceph-volume simple scan /var/lib/ceph/osd/ceph-0
 stderr: lsblk: /var/lib/ceph/osd/ceph-0: not a block device
 stderr: lsblk: /var/lib/ceph/osd/ceph-0: not a block device
 stderr: Unknown device, --name=, --path=, or absolute path in /dev/ or /sys expected.
Running command: /usr/sbin/cryptsetup status /dev/sdb1
--> OSD 0 got scanned and metadata persisted to file: /etc/ceph/osd/0-0c9917f7-fce8-42aa-bdec-8c2cf2d536ba.json
--> To take over management of this scanned OSD, and disable ceph-disk and udev, run:
-->     ceph-volume simple activate 0 0c9917f7-fce8-42aa-bdec-8c2cf2d536ba
```

对任何其他对象存储设备重复上述步骤。

c. 激活该设备：

语法

```
ceph-volume simple activate OSD_ID UUID
```

-

将 *OSD_ID* 替换为 OSD ID，*UUID* 替换为之前在扫描输出中输出的 UUID。

示例

```
[root@mon ~]# ceph-volume simple activate 0 0c9917f7-fce8-42aa-bdec-8c2cf2d536ba
Running command: /usr/bin/ln -snf /dev/sdb2 /var/lib/ceph/osd/ceph-0/journal
Running command: /usr/bin/chown -R ceph:ceph /dev/sdb2
Running command: /usr/bin/systemctl enable ceph-volume@simple-0-0c9917f7-fce8-
42aa-bdec-8c2cf2d536ba
 stderr: Created symlink /etc/systemd/system/multi-user.target.wants/ceph-
volume@simple-0-0c9917f7-fce8-42aa-bdec-8c2cf2d536ba.service →
/usr/lib/systemd/system/ceph-volume@.service.
Running command: /usr/bin/ln -sf /dev/null /etc/systemd/system/ceph-disk@.service
--> All ceph-disk systemd units have been disabled to prevent OSDs getting triggered by
UDEV events
Running command: /usr/bin/systemctl enable --runtime ceph-osd@0
 stderr: Created symlink /run/systemd/system/ceph-osd.target.wants/ceph-
osd@0.service → /usr/lib/systemd/system/ceph-osd@.service.
Running command: /usr/bin/systemctl start ceph-osd@0
--> Successfully activated OSD 0 with FSID 0c9917f7-fce8-42aa-bdec-8c2cf2d536ba
```

对任何其他对象存储设备重复上述步骤。

20. 可选：如果您的 OSD 是使用 ceph-volume 创建的，并且您没有完成上一步，请立即启动 OSD 服务：

```
[root@mon ~]# systemctl start ceph-osd.target
```

21. 激活 OSD：

BlueStore

```
[root@mon ~]# ceph-volume lvm activate --all
```

22. **验证 OSDs 为 up 和 in，它们处于 active+clean 状态。**

```
[root@mon ~]# ceph -s
```

在 *services:* 下的 *osd:* 行中，确定所有 OSDs 都为 up 和 in：

示例

```
osd: 3 osds: 3 up (since 8s), 3 in (since 3M)
```

23. **在所有 OSD 节点上重复上述步骤，直到它们都已升级。**

24. **如果从 Red Hat Ceph Storage 3 升级，则不允许预先Nautilus OSD 并启用只有 Nautilus 的功能：**

```
[root@mon ~]# ceph osd require-osd-release nautilus
```

注意

未能执行此步骤会导致 OSD 在启用 msgrv2 后无法进行通信。

25. **从 Red Hat Ceph Storage 3 升级到 Red Hat Ceph Storage 4 后，即可执行以下步骤：**

a. **启用消息传递 v2 协议 msgr2：**

```
[root@mon ~]# ceph mon enable-msgr2
```

这将指示绑定到 6789 的旧默认端口的所有 Ceph 监控器也绑定到 3300 的新端口。

b. 在每个节点上，将 ceph.conf 文件中的任何配置选项导入到存储集群的配置数据库中：

示例

```
[root@mon ~]# ceph config assimilate-conf -i /etc/ceph/ceph.conf
```

注意

当将配置填充到监控器时，例如，如果您为同一组选项设置了不同的配置值，则最终结果会取决于文件完成的顺序。

c. 检查存储集群的配置数据库：

示例

```
[root@mon ~]# ceph config dump
```

其它资源

- 如需更多信息，请参阅安装指南中的手动升级 Red Hat Ceph Storage 集群和操作系统。
- 如需更多信息，请参阅 从 Red Hat Enterprise Linux 7 升级到 Red Hat Enterprise Linux 8。

8.4. 手动升级 CEPH 对象网关节点及其操作系统

作为系统管理员，您可以手动将 Red Hat Ceph Storage 集群节点上的 Ceph Object Gateway (RGW) 软件和 Red Hat Enterprise Linux 操作系统同时升级到新的主版本。

重要

应当对 Ceph 集群中的每一 RGW 节点执行此步骤，但一次仅针对一个 RGW 节点执行此步骤。在继续下一节点之前，确保当前升级的 RGW 已恢复正常操作，以防止任何客户端访问问题。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 节点正在运行 Red Hat Enterprise Linux 7 7.9。
- 节点使用 Red Hat Ceph Storage 版本 3.3z6 或 4.1
- 访问 Red Hat Enterprise Linux 8.3 的安装源。

流程

1. 停止 Ceph 对象网关服务：

```
# systemctl stop ceph-radosgw.target
```

2. 如果使用 Red Hat Ceph Storage 3，请禁用 Red Hat Ceph Storage 3 工具存储库：

```
# subscription-manager repos --disable=rhel-7-server-rhceph-3-tools-rpms
```

3. 如果使用 Red Hat Ceph Storage 4，请禁用 Red Hat Ceph Storage 4 工具存储库：

```
# subscription-manager repos --disable=rhel-7-server-rhceph-4-tools-rpms
```

4. 安装 leapp 实用程序。请参阅从 Red Hat Enterprise Linux 7 升级到 Red Hat Enterprise Linux 8。

5. 通过 **leapp preupgrade** 检查运行。请参阅从命令行评估可升级性。

6. 在 **/etc/ssh/sshd_config** 中设置 **PermitRootLogin yes**。

7. 重启 **OpenSSH SSH** 守护进程：

```
# systemctl restart sshd.service
```

8. 从 **Linux** 内核中删除 **iSCSI** 模块：

```
# modprobe -r iscsi
```

9. 执行 **从 RHEL 7 升级到 RHEL 8** 的内容，以执行升级。

10. 重新引导节点。

11. 为 **Red Hat Enterprise Linux 8** 启用 **Red Hat Ceph Storage 4** 的工具存储库。

```
# subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
```

12. 安装 **ceph-radosgw** 软件包：

```
# dnf install ceph-radosgw
```

13. 可选：安装在此节点上并置的任何 **Ceph** 服务的软件包。如果需要，启用额外的 **Ceph** 存储库。

14. 可选：安装其他 **Ceph** 服务需要的 **leveldb** 软件包。

```
# dnf install leveldb
```

15. 从尚未升级的节点或已经恢复这些文件的节点恢复 **ceph-client-admin.keyring** 和 **ceph.conf** 文件。

16. 启动 **RGW** 服务：

```
# systemctl start ceph-radosgw.target
```

17. 将任何现有的 **CRUSH bucket** 切换到最新的 **bucket** 类型 **straw2**。

```
# ceph osd getcrushmap -o backup-crushmap
# ceph osd crush set-all-straw-buckets-to-straw2
```

18. 验证守护进程是否活跃：

```
# ceph -s
```

在 ***services:*** 下有一个 ***rgw:*** 行。

示例

```
rgw: 1 daemon active (jb-ceph4-rgw.rgw0)
```

19. 在所有 **Ceph** 对象网关节点上重复上述步骤，直到它们都已升级。

其它资源

- 如需更多信息，请参阅安装指南中的手动升级 **Red Hat Ceph Storage** 集群和操作系统。
- 如需更多信息，请参阅 从 **Red Hat Enterprise Linux 7** 升级到 **Red Hat Enterprise Linux 8**。

8.5. 手动升级 CEPH 控制面板节点及其操作系统

作为系统管理员，您可以手动将 Red Hat Ceph Storage 集群节点上的 Ceph Dashboard 软件和 Red Hat Enterprise Linux 操作系统同时升级到新的主版本。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 该节点正在运行 Red Hat Enterprise Linux 7.9。
- 该节点正在运行 Red Hat Ceph Storage 版本 3.3z6 或 4.1
- 访问 Red Hat Enterprise Linux 8.3 的安装源。

流程

1. 从集群卸载现有的仪表板。

   a. 进入 /usr/share/cephmetrics-ansible 目录：

   ```
   # cd /usr/share/cephmetrics-ansible
   ```

   b. 运行 purge.yml Ansible playbook:

   ```
   # ansible-playbook -v purge.yml
   ```

2. 如果使用 Red Hat Ceph Storage 3，请禁用 Red Hat Ceph Storage 3 工具存储库：

   ```
   # subscription-manager repos --disable=rhel-7-server-rhceph-3-tools-rpms
   ```

3. 如果使用 Red Hat Ceph Storage 4，请禁用 Red Hat Ceph Storage 4 工具存储库：

   ```
   # subscription-manager repos --disable=rhel-7-server-rhceph-4-tools-rpms
   ```

4. 安装 leapp 实用程序。请参阅从 Red Hat Enterprise Linux 7 升级到 Red Hat Enterprise

Linux 8。

5. 通过 leapp 预升级检查运行。请参阅从命令行评估可升级性。

6. 在 /etc/ssh/sshd_config 中设置 PermitRootLogin yes。

7. 重启 OpenSSH SSH 守护进程：

```
# systemctl restart sshd.service
```

8. 从 Linux 内核中删除 iSCSI 模块：

```
# modprobe -r iscsi
```

9. 执行 从 RHEL 7 升级到 RHEL 8 的内容，以执行升级。

10. 重新引导节点。

11. 为 Red Hat Enterprise Linux 8 启用 Red Hat Ceph Storage 4 的工具存储库：

```
# subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
```

12. 启用 Ansible 存储库：

```
# subscription-manager repos --enable=ansible-2.9-for-rhel-8-x86_64-rpms
```

13. 配置 ceph-ansible 以管理集群。它将安装仪表板。按照使用 Ansible 安装Red Hat Ceph Storage 中的说明，包括前提条件。

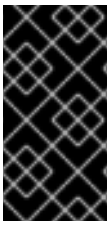
14. 作为上述流程的一部分运行 ansible-playbook site.yml 后，会输出仪表板的 URL。如需有关查找 URL 和访问仪表板的更多信息，请参阅 控制面板指南中使用 Ansible 安装仪表板。

其它资源

- 如需更多信息，请参阅安装指南中的手动升级 Red Hat Ceph Storage 集群和操作系统。
- 如需更多信息，请参阅 从 Red Hat Enterprise Linux 7 升级到 Red Hat Enterprise Linux 8。
- 如需更多信息，请参阅 控制面板指南中的Ansible 安装仪表板。

8.6. 手动升级 CEPH ANSIBLE 节点并重新配置设置

将 Red Hat Ceph Storage 集群节点上的 Ceph Ansible 软件和 Red Hat Enterprise Linux 操作系统手动升级到新的主要版本。除非另有指定，否则此流程适用于裸机和容器部署。



重要

在 Ceph Ansible 节点上升级 hostOS 之前，先对 group_vars 和 hosts 文件进行备份。在重新配置 Ceph Ansible 节点之前，使用创建的备份。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 该节点正在运行 Red Hat Enterprise Linux 7.9。
- 该节点正在运行 Red Hat Ceph Storage 版本 3.3z6 或 4.1
- 访问 Red Hat Enterprise Linux 8.3 的安装源。

流程

1. 为 Red Hat Enterprise Linux 8 启用 Red Hat Ceph Storage 4 的工具存储库：

```
[root@dashboard ~]# subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
```

2. 启用 Ansible 存储库：

```
[root@dashboard ~]# subscription-manager repos --enable=ansible-2.9-for-rhel-8-x86_64-rpms
```

3. 配置 ceph-ansible 以管理存储集群。它将安装仪表板。按照使用 Ansible 安装Red Hat Ceph Storage 中的说明，包括前提条件。

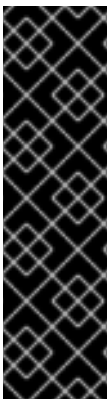
4. 作为上述流程的一部分运行 ansible-playbook site.yml 后，会输出仪表板的 URL。如需有关查找 URL 和访问仪表板的更多信息，请参阅 控制面板指南中使用 Ansible 安装仪表板。

其它资源

- 如需更多信息，请参阅安装指南中的手动升级 Red Hat Ceph Storage 集群和操作系统。
- 如需更多信息，请参阅 从 Red Hat Enterprise Linux 7 升级到 Red Hat Enterprise Linux 8。
- 如需更多信息，请参阅 控制面板指南中的Ansible 安装仪表板。

8.7. 手动升级 CEPH 文件系统元数据服务器节点及其操作系统

您可以手动将 Red Hat Ceph Storage 集群中的 Ceph 文件系统 (CephFS) 元数据服务器 (MDS) 软件同时升级到新的主版本。



重要

在升级存储集群前，请将活跃 MDS 的数量减少为每个文件系统一个。这消除了多个 MDS 之间可能存在的版本冲突。另外，在升级前关闭所有待机节点。

这是因为 MDS 集群没有内置的版本或文件系统标志。如果没有这些功能，多个 MDS 可能会使用不同版本的 MDS 软件进行通信，并可能导致断言或其他故障发生。

先决条件

- 正在运行的 Red Hat Ceph Storage 集群。
- 节点正在运行 Red Hat Enterprise Linux 7.9。
- 节点使用 Red Hat Ceph Storage 版本 3.3z6 或 4.1。
- 访问 Red Hat Enterprise Linux 8.3 的安装源。
- 对存储集群中所有节点的根级别访问权限。

重要

底层 XFS 文件系统必须格式化为支持 ftype=1 或 d_type。运行 xfs_info /var 命令以确保 ftype 设置为 1。如果 ftype 的值不是 1，请附加新磁盘或创建卷。在此新设备之上，创建新的 XFS 文件系统并将其挂载到 /var/lib/containers。

从 Red Hat Enterprise Linux 8 开始，mkfs.xfs 默认启用 ftype=1。

流程

1. 将活跃 MDS 的数量减少到 1：

语法

```
ceph fs set FILE_SYSTEM_NAME max_mds 1
```

示例

```
[root@mds ~]# ceph fs set fs1 max_mds 1
```

2. 等待集群停止所有 MDS 等级。当所有 MDS 停止后，仅排名 0 才处于活动状态。剩余的操作应处于待机模式。检查文件系统的状态：

```
[root@mds ~]# ceph status
```

3. 使用 systemctl 关闭所有备用 MDS：

```
[root@mds ~]# systemctl stop ceph-mds.target
```

4. 确认只有一个 MDS 是在线的，并且它已在文件系统中排名为 0：

```
[root@mds ~]# ceph status
```

5. 为操作系统版本禁用工具存储库：

   a. 如果您要从 RHEL 7 上的 Red Hat Ceph Storage 3 升级，请禁用 Red Hat Ceph Storage 3 工具存储库：

   ```
   [root@mds ~]# subscription-manager repos --disable=rhel-7-server-rhceph-3-tools-rpms
   ```

   b. 如果您使用 Red Hat Ceph Storage 4，请禁用 Red Hat Ceph Storage 4 工具存储库：

   ```
   [root@mds ~]# subscription-manager repos --disable=rhel-7-server-rhceph-4-tools-rpms
   ```

6. 安装 leapp 实用程序。有关 leapp 的详情，请参考从 Red Hat Enterprise Linux 7 升级到 Red Hat Enterprise Linux 8。

7. 通过 leapp 预升级检查运行。如需更多信息，请参阅从命令行评估可升级性。

8. 编辑 /etc/ssh/sshd_config，并将 PermitRootLogin 设置为 yes。

9. **重启 OpenSSH SSH 守护进程：**

```
[root@mds ~]# systemctl restart sshd.service
```

10. **从 Linux 内核中删除 iSCSI 模块：**

```
[root@mds ~]# modprobe -r iscsi
```

11. **执行升级。请参阅执行从 RHEL 7 升级到 RHEL 8。**

12. **重新引导 MDS 节点。**

13. **为 Red Hat Enterprise Linux 8 启用 Red Hat Ceph Storage 4 的工具仓库：**

```
[root@mds ~]# subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
```

14. **安装 ceph-mds 软件包：**

```
[root@mds ~]# dnf install ceph-mds -y
```

15. **可选：安装在此节点上并置的任何 Ceph 服务的软件包。如果需要，启用额外的 Ceph 存储库。**

16. **可选：安装其他 Ceph 服务需要的 leveldb 软件包：**

```
[root@mds ~]# dnf install leveldb
```

17. **从尚未升级的节点或已经恢复这些文件的节点恢复 ceph-client-admin.keyring 和 ceph.conf 文件。**

18. **将任何现有的 CRUSH bucket 切换到最新的 bucket 类型 straw2。**

```
# ceph osd getcrushmap -o backup-crushmap
# ceph osd crush set-all-straw-buckets-to-straw2
```

19. 启动 **MDS** 服务：

```
[root@mds ~]# systemctl restart ceph-mds.target
```

20. 验证守护进程是否活跃：

```
[root@mds ~]# ceph -s
```

21. 为待机守护进程跟踪相同的进程。

22. 当您完成重启所有待机 **MDS** 后，请恢复集群中的 **max_mds** 的值：

语法

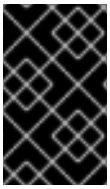
```
ceph fs set FILE_SYSTEM_NAME max_mds ORIGINAL_VALUE
```

示例

```
[root@mds ~]# ceph fs set fs1 max_mds 5
```

8.8. 从 OSD 节点上的操作系统升级失败中恢复

作为系统管理员，如果您在使用手动升级 Ceph OSD 节点及其操作系统的步骤时失败，您可以按照以下步骤从故障中恢复：在该过程中，您将在节点上全新安装 **Red Hat Enterprise Linux 8.4**，并且仍然能够恢复 **OSD**，而不必回填数据，除了写入到它们已停机的 **OSD** 外。



重要

不要触动支持 OSD 或对应的 wal.db 或 block.db 数据库的介质。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 升级失败的 OSD 节点。
- 访问 Red Hat Enterprise Linux 8.4 安装源。

流程

1. 在失败节点中执行 Red Hat Enterprise Linux 8.4 标准安装并启用 Red Hat Enterprise Linux 软件仓库。
   - 执行标准 RHEL 安装
2. 为 Red Hat Enterprise Linux 8 启用 Red Hat Ceph Storage 4 的软件仓库。
   a. 启用工具存储库：

   ```
   # subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
   ```

   b. 启用 osd 存储库：

   ```
   # subscription-manager repos --enable=rhceph-4-osd-for-rhel-8-x86_64-rpms
   ```
3. 安装 ceph-osd 软件包：

   ```
   # dnf install ceph-osd
   ```

4. 将 ceph.conf 文件从尚未升级的节点或已经恢复这些文件的节点恢复到 /etc/ceph。

5. 启动 OSD 服务：

```
# systemctl start ceph-osd.target
```

6. 激活对象存储设备：

```
ceph-volume lvm activate --all
```

7. 观察 OSD 的恢复和集群回填写入恢复的 OSD：

```
# ceph -w
```

监控输出，直到所有 PG 都处于 active+clean 状态。

其它资源

- 如需更多信息，请参阅安装指南中的手动升级 Red Hat Ceph Storage 集群和操作系统。
- 如需更多信息，请参阅 从 Red Hat Enterprise Linux 7 升级到 Red Hat Enterprise Linux 8。

8.9. 其它资源

- 如果您不需要将操作系统升级到新的主版本，请参阅升级 Red Hat Ceph Storage 集群。

第 9 章 接下来该怎么办？

这仅仅是 Red Hat Ceph Storage 为帮助您满足现代数据中心富有挑战性的存储需求所它可以起到的作用的开始。以下是有关各种主题的更多信息的链接：

- 基准测试性能和访问性能计数器，请参见 Red Hat Ceph Storage 4 管理指南中的*基准测试性能*章节。
- 快照的创建和管理，请参见 Red Hat Ceph Storage 4 块设备指南中的*快照*一章。
- 扩展 Red Hat Ceph Storage，请参阅 Red Hat Ceph Storage 4 操作指南中的*管理存储集群大小*一章。
- 镜像 Ceph 块设备，请参见 Red Hat Ceph Storage 4 块设备指南中的*块设备镜像*一章。
- 流程管理，请参见 Red Hat Ceph Storage 4 管理指南中的*进程管理*一章。
- 对于可微调参数，请参阅 Red Hat Ceph Storage 4 的*配置指南*。
- 将 Ceph 用作 OpenStack 的后端存储，请参见 Red Hat OpenStack Platform 存储指南中的后端章节。
- 利用 Ceph 控制面板监控 Red Hat Ceph Storage 集群的健康和容量。如需了解更多详细信息，请参阅 *控制面板指南*。

附录 A. 故障排除

A.1. ANSIBLE 停止安装，因为它检测到的设备比预期少

Ansible 自动化应用程序停止安装过程并返回以下错误：

```
- name: fix partitions gpt header or labels of the osd disks (autodiscover disks)
  shell: "sgdisk --zap-all --clear --mbrtogpt -- '/dev/{{ item.0.item.key }}' || sgdisk --zap-all --clear --
mbrtogpt -- '/dev/{{ item.0.item.key }}'"
  with_together:
    - "{{ osd_partition_status_results.results }}"
    - "{{ ansible_devices }}"
  changed_when: false
  when:
    - ansible_devices is defined
    - item.0.item.value.removable == "0"
    - item.0.item.value.partitions|count == 0
    - item.0.rc != 0
```

这意味着：

当 /usr/share/ceph-ansible/group_vars/osds.yml 文件中的 osd_auto_discovery 参数设置为 true 时，Ansible 会自动检测并配置所有可用的设备。在这一过程中，Ansible 期望所有 OSD 都使用相同的设备。设备按照 Ansible 检测到的名称的顺序获得它们的名称。如果其中一个设备在其中一个 OSD 上失败，Ansible 无法检测到失败的设备并停止整个安装过程。

示例情况：

1. 三个 OSD 节点（host1、host2、host3）使用 /dev/sdb、/dev/sdc 和 dev/sdd 磁盘。
2. 在 host2 上，/dev/sdc 磁盘失败并被删除。
3. 下一次重启后，Ansible 无法检测已移除的 /dev/sdc 磁盘，并且希望只有两个磁盘将用于 host2，即/dev/sdb 和 /dev/sdc （以前为 /dev/sdd）。
4. Ansible 将停止安装过程并返回上述错误消息。

解决此问题：

在 /etc/ansible/hosts 文件中，指定带有故障磁盘的 OSD 节点使用的设备（上面的示例中为host2

）：

```
[osds]
host1
host2 devices="[ '/dev/sdb', '/dev/sdc' ]"
host3
```

详情请查看 *第 5 章 使用 Ansible 安装 Red Hat Ceph Storage*。

附录 B. 使用命令行界面安装 CEPH 软件

作为存储管理员，您可以选择手动安装 Red Hat Ceph Storage 软件的各种组件。

B.1. 安装 CEPH 命令行界面

Ceph 命令行界面 (CLI) 让管理员可以执行 Ceph 管理命令。CLI 由 ceph-common 软件包提供，包括以下实用程序：

- ceph
- ceph-authtool
- ceph-dencoder
- rados

先决条件

- 正在运行的 Ceph 存储集群，最好处于 active + clean 状态。

流程

1. 在客户端节点上，启用 Red Hat Ceph Storage 4 Tools 存储库：

```
[root@gateway ~]# subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-rpms
```

2. 在客户端节点上安装 ceph-common 软件包：

```
# yum install ceph-common
```

3. 从初始监控节点，复制 Ceph 配置文件，本例中为 ceph.conf，以及管理密钥环到客户端节点：

语法

```
# scp /etc/ceph/ceph.conf <user_name>@<client_host_name>:/etc/ceph/
# scp /etc/ceph/ceph.client.admin.keyring <user_name>@<client_host_name:/etc/ceph/
```

示例

```
# scp /etc/ceph/ceph.conf root@node1:/etc/ceph/
# scp /etc/ceph/ceph.client.admin.keyring root@node1:/etc/ceph/
```

将 **<client_host_name>** 替换为客户端节点的主机名。

B.2. 手动安装 RED HAT CEPH STORAGE

重要

红帽不支持或测试手动部署的集群的升级。因此，红帽建议使用 **Ansible** 来使用 **Red Hat Ceph Storage 4** 部署新集群。详情请查看 *第 5 章 使用 Ansible 安装 Red Hat Ceph Storage*。

您可以使用 **Yum** 等命令行实用程序升级手动部署的集群，但红帽不支持或测试这种方法。

所有 **Ceph** 集群需要至少一个 **monitor**，并且至少与集群中存储的对象副本数量相同。红帽建议在生产环境中使用三个监视器，至少三个对象存储设备 **(OSD)**。

启动初始监控器是部署 **Ceph** 存储集群的第一步。**Ceph** 监控部署还为整个集群设置重要标准，例如：

- 池的副本数
- 每个 OSD 的 PG 数量
- 心跳间隔
- 任何身份验证要求

默认情况下会设置其中大多数值，因此为生产环境设置集群时了解这些值非常有用。

使用命令行界面安装 Ceph 存储集群涉及以下步骤：

- 引导初始监控节点
- 添加对象存储设备 (OSD)节点

监控 Bootstrapping

引导 monitor 和扩展 Ceph 存储集群需要以下数据：

唯一标识符

文件系统标识符 (fsid) 是集群的唯一标识符。fsid 最初在 Ceph 存储集群主要用于 Ceph 文件系统时使用。Ceph 现在也支持原生接口、块设备和对象存储网关接口，因此 fsid 可能会有一些问题。

Monitor 名称

集群中的每一个 Monitor 实例都有唯一的名称。在常见做法中，Ceph monitor 名称是节点名称。红帽建议每个节点一个 Ceph 监控器，而不与 Ceph 监控守护进程共同定位 Ceph OSD 守护进程。要获得较短的节点名称，请使用 hostname -s 命令。

Monitor Map

启动初始 Monitor 要求您生成 Monitor Map。Monitor map 需要：

- 文件系统识别符 (fsid)
- 使用集群名称或 ceph 的默认集群名称
- 至少一个主机名及其 IP 地址。

监控密钥环

Monitor 使用 secret 密钥相互通信。您必须使用 Monitor secret 密钥生成密钥环，并在引导初始 Monitor 时提供密钥环。

管理员密钥环

要使用 ceph 命令行界面实用程序，请创建 client.admin 用户并生成其密钥环。此外，您必须将 client.admin 用户添加到 monitor 密钥环中。

强制要求不表示创建 Ceph 配置文件。但是，作为一种最佳实践，红帽建议创建一个 Ceph 配置文件并使用 fsid 填充它的数据，mon initial members 和 mon host 是最小设置。

您还可以在运行时获取和设置所有 Monitor 设置。但是，Ceph 配置文件可能仅包含覆盖默认值的设置。当您向 Ceph 配置文件添加设置时，这些设置将覆盖默认设置。在 Ceph 配置文件中维护这些设置可以更加轻松地维护集群。

要引导初始 Monitor，请执行以下步骤：

1. 启用 Red Hat Ceph Storage 4 Monitor 存储库：

```
[root@monitor ~]# subscription-manager repos --enable=rhceph-4-mon-for-rhel-8-x86_64-rpms
```

2. 在初始监控节点上，以 root 用户身份安装 ceph-mon 软件包：

```
# yum install ceph-mon
```

3. 以 root 用户身份，在 /etc/ceph/ 目录中创建 Ceph 配置文件。

```
# touch /etc/ceph/ceph.conf
```

4. **以 root 用户身份，为集群生成唯一标识符，并将唯一标识符添加到 Ceph 配置文件的 [global] 部分：**

```
# echo "[global]" > /etc/ceph/ceph.conf
# echo "fsid = `uuidgen`" >> /etc/ceph/ceph.conf
```

5. **查看当前的 Ceph 配置文件：**

```
$ cat /etc/ceph/ceph.conf
[global]
fsid = a7f64266-0894-4f1e-a635-d0aeaca0e993
```

6. **以 root 用户身份，将初始 monitor 添加到 Ceph 配置文件：**

语法

```
# echo "mon initial members = <monitor_host_name>[,<monitor_host_name>]" >>
/etc/ceph/ceph.conf
```

示例

```
# echo "mon initial members = node1" >> /etc/ceph/ceph.conf
```

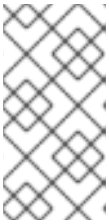
7. **以 root 用户身份，将初始 monitor 的 IP 地址添加到 Ceph 配置文件：**

语法

```
# echo "mon host = <ip-address>[,<ip-address>]" >> /etc/ceph/ceph.conf
```

示例

```
# echo "mon host = 192.168.0.120" >> /etc/ceph/ceph.conf
```



注意

要使用 IPv6 地址，请将 ms bind ipv6 选项设置为 true。详情请参阅 Red Hat Ceph Storage 4 配置指南中的 绑定"一节。

8. **以 root 用户身份，为集群创建密钥环并生成 monitor secret 密钥：**

```
# ceph-authtool --create-keyring /tmp/ceph.mon.keyring --gen-key -n mon. --cap mon 'allow *'
creating /tmp/ceph.mon.keyring
```

9. **以 root 身份生成管理员密钥环，生成 ceph.client.admin.keyring 用户，并将该用户添加到密钥环中：**

语法

```
# ceph-authtool --create-keyring /etc/ceph/ceph.client.admin.keyring --gen-key -n
client.admin --set-uid=0 --cap mon '<capabilites>' --cap osd '<capabilites>' --cap mds
'<capabilites>'
```

示例

```
# ceph-authtool --create-keyring /etc/ceph/ceph.client.admin.keyring --gen-key -n
client.admin --set-uid=0 --cap mon 'allow *' --cap osd 'allow *' --cap mds 'allow'
creating /etc/ceph/ceph.client.admin.keyring
```

10. **以 root 用户身份，将 ceph.client.admin.keyring 密钥添加到 ceph.mon.keyring 中：**

```
# ceph-authtool /tmp/ceph.mon.keyring --import-keyring /etc/ceph/ceph.client.admin.keyring
importing contents of /etc/ceph/ceph.client.admin.keyring into /tmp/ceph.mon.keyring
```

11. **生成 Monitor map。使用初始 monitor 的节点名称、IP 地址和 fsid 指定，并将其保存为 /tmp/monmap：**

语法

```
$ monmaptool --create --add <monitor_host_name> <ip-address> --fsid <uuid>
/tmp/monmap
```

示例

```
$ monmaptool --create --add node1 192.168.0.120 --fsid a7f64266-0894-4f1e-a635-
d0aeaca0e993 /tmp/monmap
monmaptool: monmap file /tmp/monmap
monmaptool: set fsid to a7f64266-0894-4f1e-a635-d0aeaca0e993
monmaptool: writing epoch 0 to /tmp/monmap (1 monitors)
```

12. **作为初始监控节点上的 root 用户，创建一个默认数据目录：**

语法

```
# mkdir /var/lib/ceph/mon/ceph-<monitor_host_name>
```

示例

```
# mkdir /var/lib/ceph/mon/ceph-node1
```

13. **以 root 用户身份，使用 monitor 映射和密钥环填充初始 monitor 守护进程：**

语法

```
# ceph-mon --mkfs -i <monitor_host_name> --monmap /tmp/monmap --keyring
/tmp/ceph.mon.keyring
```

示例

```
# ceph-mon --mkfs -i node1 --monmap /tmp/monmap --keyring /tmp/ceph.mon.keyring
ceph-mon: set fsid to a7f64266-0894-4f1e-a635-d0aeaca0e993
ceph-mon: created monfs at /var/lib/ceph/mon/ceph-node1 for mon.node1
```

14. **查看当前的 Ceph 配置文件：**

```
# cat /etc/ceph/ceph.conf
[global]
fsid = a7f64266-0894-4f1e-a635-d0aeaca0e993
mon_initial_members = node1
mon_host = 192.168.0.120
```

有关各种 Ceph 配置设置的更多详细信息，请参见 Red Hat Ceph Storage 4 配置指南 。以

下 Ceph 配置文件示例列出了一些最常见的配置设置：

示例

```
[global]
fsid = <cluster-id>
mon initial members = <monitor_host_name>[, <monitor_host_name>]
mon host = <ip-address>[, <ip-address>]
public network = <network>[, <network>]
cluster network = <network>[, <network>]
auth cluster required = cephx
auth service required = cephx
auth client required = cephx
osd journal size = <n>
osd pool default size = <n>  # Write an object n times.
osd pool default min size = <n> # Allow writing n copy in a degraded state.
osd pool default pg num = <n>
osd pool default pgp num = <n>
osd crush chooseleaf type = <n>
```

15. 以 root 用户身份，创建 done 文件：

语法

```
# touch /var/lib/ceph/mon/ceph-<monitor_host_name>/done
```

示例

```
# touch /var/lib/ceph/mon/ceph-node1/done
```

16.

以 root 用户身份，更新新创建的目录和文件的所有者和组权限：

语法

```
# chown -R <owner>:<group> <path_to_directory>
```

示例

```
# chown -R ceph:ceph /var/lib/ceph/mon
# chown -R ceph:ceph /var/log/ceph
# chown -R ceph:ceph /var/run/ceph
# chown ceph:ceph /etc/ceph/ceph.client.admin.keyring
# chown ceph:ceph /etc/ceph/ceph.conf
# chown ceph:ceph /etc/ceph/rbdmap
```

注意

如果 Ceph 监控节点与 OpenStack 控制器节点在一起，则 Glance 和 Cinder 密钥环文件必须分别归 glance 和 cinder 所有。例如：

```
# ls -l /etc/ceph/
...
-rw-------.  1 glance glance      64 <date> ceph.client.glance.keyring
-rw-------.  1 cinder cinder      64 <date> ceph.client.cinder.keyring
...
```

17.

以 root 用户身份，在初始监控节点上启动并启用 ceph-mon 进程：

语法

```
# systemctl enable ceph-mon.target
# systemctl enable ceph-mon@<monitor_host_name>
# systemctl start ceph-mon@<monitor_host_name>
```

示例

```
# systemctl enable ceph-mon.target
# systemctl enable ceph-mon@node1
# systemctl start ceph-mon@node1
```

18. **以 root 用户身份，验证 monitor 守护进程是否正在运行：**

语法

```
# systemctl status ceph-mon@<monitor_host_name>
```

示例

```
# systemctl status ceph-mon@node1
● ceph-mon@node1.service - Ceph cluster monitor daemon
   Loaded: loaded (/usr/lib/systemd/system/ceph-mon@.service; enabled; vendor preset:
disabled)
   Active: active (running) since Wed 2018-06-27 11:31:30 PDT; 5min ago
 Main PID: 1017 (ceph-mon)
   CGroup: /system.slice/system-ceph\x2dmon.slice/ceph-mon@node1.service
           └─1017 /usr/bin/ceph-mon -f --cluster ceph --id node1 --setuser ceph --setgroup ceph

Jun 27 11:31:30 node1 systemd[1]: Started Ceph cluster monitor daemon.
Jun 27 11:31:30 node1 systemd[1]: Starting Ceph cluster monitor daemon...
```

要将更多 Red Hat Ceph Storage Monitor 添加到存储集群中，请参阅 Red Hat Ceph Storage 4 管理指南中的添加 Monitor 部分。

OSD Bootstrapping

运行初始监控器后，您可以开始添加对象存储设备 (OSD)。直到有足够的 OSD 来处理对象的副本数时，您的集群才会达到 active + clean 状态。

对象的默认副本数为三个。至少需要三个 OSD 节点：但是，如果您只需要一个对象的两个副本，因此仅添加两个 OSD 节点，然后更新 Ceph 配置文件中的 osd pool default size 和 osd pool default min size 设置。

如需了解更多详细信息，请参阅 Red Hat Ceph Storage 4 *配置指南*中的*OSD 配置参考*一节。

在引导初始监控器后，集群具有默认的 CRUSH map。但是，CRUSH map 没有任何 Ceph OSD 守护进程映射到 Ceph 节点。

要添加 OSD 到集群并更新默认的 CRUSH map，请在每个 OSD 节点上执行以下内容：

1. 启用 Red Hat Ceph Storage 4 OSD 存储库：

```
[root@osd ~]# subscription-manager repos --enable=rhceph-4-osd-for-rhel-8-x86_64-rpms
```

2. 以 root 用户身份，在 Ceph OSD 节点上安装 ceph-osd 软件包：

```
# yum install ceph-osd
```

3. 将 Ceph 配置文件和管理密钥环文件从初始 Monitor 节点复制到 OSD 节点：

语法

```
# scp <user_name>@<monitor_host_name>:<path_on_remote_system>
<path_to_local_file>
```

示例

```
# scp root@node1:/etc/ceph/ceph.conf /etc/ceph
# scp root@node1:/etc/ceph/ceph.client.admin.keyring /etc/ceph
```

4. 为 **OSD 生成通用唯一标识符 (UUID)：**

```
$ uuidgen
b367c360-b364-4b1d-8fc6-09408a9cda7a
```

5. **以 root 用户身份，创建 OSD 实例：**

语法

```
# ceph osd create <uuid> [<osd_id>]
```

示例

```
# ceph osd create b367c360-b364-4b1d-8fc6-09408a9cda7a
0
```

注意

此命令输出后续步骤所需的 OSD 编号标识符。

6. **以 root 用户身份，为新 OSD 创建默认目录：**

语法

```
# mkdir /var/lib/ceph/osd/ceph-<osd_id>
```

示例

```
# mkdir /var/lib/ceph/osd/ceph-0
```

7. **以 root 用户身份，准备好将驱动器用作 OSD，并将它挂载到您刚才创建的目录中。为 Ceph 数据和日志创建一个分区。日志和数据分区可以位于同一磁盘上。这个示例使用 15 GB 磁盘：**

语法

```
# parted <path_to_disk> mklabel gpt
# parted <path_to_disk> mkpart primary 1 10000
# mkfs -t <fstype> <path_to_partition>
# mount -o noatime <path_to_partition> /var/lib/ceph/osd/ceph-<osd_id>
# echo "<path_to_partition>  /var/lib/ceph/osd/ceph-<osd_id>   xfs defaults,noatime 1 2" >>
/etc/fstab
```

示例

```
# parted /dev/sdb mklabel gpt
# parted /dev/sdb mkpart primary 1 10000
# parted /dev/sdb mkpart primary 10001 15000
```

```
# mkfs -t xfs /dev/sdb1
# mount -o noatime /dev/sdb1 /var/lib/ceph/osd/ceph-0
# echo "/dev/sdb1 /var/lib/ceph/osd/ceph-0  xfs defaults,noatime 1 2" >> /etc/fstab
```

8. **以 root 用户身份，初始化 OSD 数据目录：**

语法

```
# ceph-osd -i <osd_id> --mkfs --mkkey --osd-uuid <uuid>
```

示例

```
# ceph-osd -i 0 --mkfs --mkkey --osd-uuid b367c360-b364-4b1d-8fc6-09408a9cda7a
... auth: error reading file: /var/lib/ceph/osd/ceph-0/keyring: can't open /var/lib/ceph/osd/ceph-0/keyring: (2) No such file or directory
... created new key in keyring /var/lib/ceph/osd/ceph-0/keyring
```

9. **以 root 身份，注册 OSD 身份验证密钥。**

语法

```
# ceph auth add osd.<osd_id> osd 'allow *' mon 'allow profile osd' -i /var/lib/ceph/osd/ceph-<osd_id>/keyring
```

示例

```
# ceph auth add osd.0 osd 'allow *' mon 'allow profile osd' -i /var/lib/ceph/osd/ceph-0/keyring
added key for osd.0
```

10. **以 root 用户身份，将 OSD 节点添加到 CRUSH map：**

语法

```
# ceph osd crush add-bucket <host_name> host
```

示例

```
# ceph osd crush add-bucket node2 host
```

11. **以 root 用户身份，将 OSD 节点放在 default CRUSH 树下：**

语法

```
# ceph osd crush move <host_name> root=default
```

示例

```
# ceph osd crush move node2 root=default
```

12. **以 root 用户身份，将 OSD 磁盘添加到 CRUSH map**

语法

```
# ceph osd crush add osd.<osd_id> <weight> [<bucket_type>=<bucket-name> ...]
```

示例

```
# ceph osd crush add osd.0 1.0 host=node2
add item id 0 name 'osd.0' weight 1 at location {host=node2} to crush map
```

注意

您也可以解译 CRUSH map，并将 OSD 添加到设备列表中。将 OSD 节点添加为 bucket，然后将设备添加为 OSD 节点中的项目，为 OSD 分配一个权重，重新编译 CRUSH map，并且设置 CRUSH map。如需了解更多详细信息，请参阅 Red Hat Ceph Storage 4 的*存储策略指南*中的编辑 CRUSH map 部分。

13. **以 root 用户身份，更新新创建的目录和文件的所有者和组权限：**

语法

```
# chown -R <owner>:<group> <path_to_directory>
```

示例

```
# chown -R ceph:ceph /var/lib/ceph/osd
# chown -R ceph:ceph /var/log/ceph
# chown -R ceph:ceph /var/run/ceph
# chown -R ceph:ceph /etc/ceph
```

14. **OSD 节点位于 Ceph 存储集群配置中。不过，OSD 守护进程为 down 和 in。新 OSD 的状态必须为 up 后才能开始接收数据。以 root 用户身份，启用并启动 OSD 过程：**

语法

```
# systemctl enable ceph-osd.target
# systemctl enable ceph-osd@<osd_id>
# systemctl start ceph-osd@<osd_id>
```

示例

```
# systemctl enable ceph-osd.target
# systemctl enable ceph-osd@0
# systemctl start ceph-osd@0
```

启动 OSD 守护进程后，它就为 up 和 in。

现在，您已启动并运行 Monitor 和一些 OSD。您可以执行以下命令来观察放置组对等点：

```
$ ceph -w
```

要查看 OSD 树，请执行以下命令：

```
$ ceph osd tree
```

示例

```
ID  WEIGHT   TYPE NAME       UP/DOWN  REWEIGHT  PRIMARY-AFFINITY
-1       2   root default
-2       2       host node2
 0       1           osd.0        up        1             1
-3       1       host node3
 1       1           osd.1        up        1             1
```

若要通过添加新 OSD 到存储集群来扩展存储容量，请参阅 Red Hat Ceph Storage 4 *管理指南*中的添加 OSD 部分。

B.3. 手动安装 CEPH MANAGER

通常，在部署 Red Hat Ceph Storage 集群时，Ansible 自动化实用程序会安装 Ceph Manager 守护进程 (ceph-mgr)。但是，如果您不使用 Ansible 管理红帽 Ceph 存储，您可以手动安装 Ceph Manager。红帽建议在同一节点上并置 Ceph 管理器和 Ceph 监控守护进程。

先决条件

- **正常工作的 Red Hat Ceph Storage 集群**
- **root 或 sudo 访问权限**
- **rhceph-4-mon-for-rhel-8-x86_64-rpms 存储库已启用**
- **如果使用防火墙，需要在公共网络上打开端口 6800-7300**

流程

在要部署 ceph-mgr 的节点上，以 root 用户身份或通过 sudo 实用程序，使用以下命令。

1. **安装 ceph-mgr 软件包：**

```
[root@node1 ~]# yum install ceph-mgr
```

2. **创建 /var/lib/ceph/mgr/ceph-*hostname*/ 目录：**

```
mkdir /var/lib/ceph/mgr/ceph-hostname
```

使用部署 ceph-mgr 守护进程的节点的主机名替换 *hostname*，例如：

```
[root@node1 ~]# mkdir /var/lib/ceph/mgr/ceph-node1
```

3. **在新创建的目录中，为 ceph-mgr 守护进程创建一个身份验证密钥：**

```
[root@node1 ~]# ceph auth get-or-create mgr.`hostname -s` mon 'allow profile mgr' osd
'allow *' mds 'allow *' -o /var/lib/ceph/mgr/ceph-node1/keyring
```

4. **将 /var/lib/ceph/mgr/ 目录的所有者和组更改为 ceph:ceph ：**

```
[root@node1 ~]# chown -R ceph:ceph /var/lib/ceph/mgr
```

5. **启用 ceph-mgr 目标：**

```
[root@node1 ~]# systemctl enable ceph-mgr.target
```

6. **启用并启动 ceph-mgr 实例：**

```
systemctl enable ceph-mgr@hostname
systemctl start ceph-mgr@hostname
```

使用部署 ceph-mgr 的节点的主机名替换 *hostname*，例如：

```
[root@node1 ~]# systemctl enable ceph-mgr@node1
[root@node1 ~]# systemctl start ceph-mgr@node1
```

7. 验证 **ceph-mgr** 守护进程是否已成功启动：

```
ceph -s
```

输出将在 **services** 部分下包括类似如下的行：

```
    mgr: node1(active)
```

8. 安装更多 **ceph-mgr** 守护进程以作为备用守护进程（如果当前活跃守护进程失败）处于活跃状态。

其他资源

- *安装 Red Hat Ceph Storage 的要求*

B.4. 手动安装 CEPH 块设备

以下步骤演示了如何安装和挂载精简调配、可调整的 Ceph 块设备。

重要

Ceph 块设备必须部署到与 Ceph 监控器和 OSD 节点上独立的节点上。在同一节点上运行内核客户端和内核服务器守护进程可能会导致内核死锁。

先决条件

- 确保执行 第 B.1 节 “安装 Ceph 命令行界面” 部分中列出的任务。
- 如果您使用 Ceph 块设备作为使用 QEMU 的虚拟机 (VM) 的后端，请增加默认的文件描述符。详情请参阅 Ceph - 虚拟机在将大量数据传输到 RBD 磁盘挂起知识库文章。

流程

1.

创建名为 client.rbd 的 Ceph 块设备用户，该用户对 OSD 节点上的文件具有完整权限 (osd 'allow rwx') 并将结果输出到密钥环文件：

```
ceph auth get-or-create client.rbd mon 'profile rbd' osd 'profile rbd pool=<pool_name>' \
-o /etc/ceph/rbd.keyring
```

将 <pool_name> 替换为您要允许 client.rbd 访问的池的名称，如 rbd ：

```
# ceph auth get-or-create \
client.rbd mon 'allow r' osd 'allow rwx pool=rbd' \
-o /etc/ceph/rbd.keyring
```

有关创建用户的更多信息，请参见 Red Hat Ceph Storage 4 *管理指南*中的*用户管理*一节。

2. 创建块设备镜像：

```
rbd create <image_name> --size <image_size> --pool <pool_name> \
--name client.rbd --keyring /etc/ceph/rbd.keyring
```

指定 <image_name>、<image_size> 和 <pool_name>，例如：

```
$ rbd create image1 --size 4G --pool rbd \
--name client.rbd --keyring /etc/ceph/rbd.keyring
```

警告

默认 Ceph 配置包括以下 Ceph 块设备功能：

- layering
- exclusive-lock
- object-map

- deep-flatten
- fast-diff

如果您使用内核 RBD (krbd) 客户端，您可能无法映射块设备镜像。

要临时解决这个问题，请禁用不支持的功能。使用以下选项之一完成此操作：

- 动态禁用不支持的功能：

```
rbd feature disable <image_name> <feature_name>
```

例如：

```
# rbd feature disable image1 object-map deep-flatten fast-diff
```

- 在 rbd create 命令中使用 --image-feature layering 选项在新创建的块设备镜像仅启用 layering。
- 在 Ceph 配置文件中禁用默认功能：

```
rbd_default_features = 1
```

这是一个已知问题，请参阅 Red Hat Ceph Storage 4 *发行说明中的已知问题* 章节。

所有这些功能适用于使用用户空间 RBD 客户端访问块设备镜像的用户。

3. 将新创建的镜像映射到块设备：

```
rbd map <image_name> --pool <pool_name>\
--name client.rbd --keyring /etc/ceph/rbd.keyring
```

例如：

```
# rbd map image1 --pool rbd --name client.rbd \
--keyring /etc/ceph/rbd.keyring
```

4. **通过创建文件系统来使用块设备：**

```
mkfs.ext4 /dev/rbd/<pool_name>/<image_name>
```

指定池名称和镜像名称，例如：

```
# mkfs.ext4 /dev/rbd/rbd/image1
```

此操作可能需要一些时间。

5. **挂载新创建的文件系统：**

```
mkdir <mount_directory>
mount /dev/rbd/<pool_name>/<image_name> <mount_directory>
```

例如：

```
# mkdir /mnt/ceph-block-device
# mount /dev/rbd/rbd/image1 /mnt/ceph-block-device
```

其它资源

- **Red Hat Ceph Storage 4** ***块设备指南*****.**

B.5. 手动安装 CEPH 对象网关

Ceph 对象网关（也称为 RADOS 网关）是在 librados API 基础上构建的对象存储接口，为应用提供 Ceph 存储集群的 RESTful 网关。

先决条件

- **正在运行的 Ceph 存储集群，最好处于 active + clean 状态。**
- **执行 第 3 章 *安装 Red Hat Ceph Storage 的要求* 中列出的任务。**

流程

1. **启用 Red Hat Ceph Storage 4 Tools 存储库：**

```
[root@gateway ~]# subscription-manager repos --enable=rhceph-4-tools-for-rhel-8-x86_64-debug-rpms
```

2. **在 Object Gateway 节点上安装 ceph-radosgw 软件包：**

```
# yum install ceph-radosgw
```

3. **在初始 monitor 节点上，执行以下步骤：**

   a. **更新 Ceph 配置文件，如下所示：**

```
[client.rgw.<obj_gw_hostname>]
host = <obj_gw_hostname>
rgw frontends = "civetweb port=80"
rgw dns name = <obj_gw_hostname>.example.com
```

   其中 <obj_gw_hostname> 是网关节点的短主机名。要查看短主机名，请使用 hostname -s 命令。

   b. **将更新的配置文件复制到新的对象网关节点和 Ceph 存储集群中的所有其他节点：**

   语法

```
# scp /etc/ceph/ceph.conf <user_name>@<target_host_name>:/etc/ceph
```

示例

```
# scp /etc/ceph/ceph.conf root@node1:/etc/ceph/
```

c. **将 ceph.client.admin.keyring 文件复制到新的对象网关节点：**

语法

```
# scp /etc/ceph/ceph.client.admin.keyring
<user_name>@<target_host_name>:/etc/ceph/
```

示例

```
# scp /etc/ceph/ceph.client.admin.keyring root@node1:/etc/ceph/
```

4. **在对象网关节点上，创建数据目录：**

```
# mkdir -p /var/lib/ceph/radosgw/ceph-rgw.`hostname -s`
```

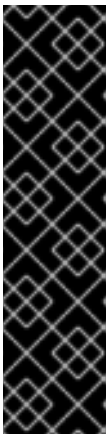
5. **在对象网关节点上，添加一个用户和密钥环来 bootstrap 对象网关：**

语法

```
# ceph auth get-or-create client.rgw.`hostname -s` osd 'allow rwx' mon 'allow rw' -o
/var/lib/ceph/radosgw/ceph-rgw.`hostname -s`/keyring
```

示例

```
# ceph auth get-or-create client.rgw.`hostname -s` osd 'allow rwx' mon 'allow rw' -o
/var/lib/ceph/radosgw/ceph-rgw.`hostname -s`/keyring
```



重要

为网关密钥提供功能时，您必须提供读取功能。但是，提供 monitor 写入功能是可选的；如果您提供此功能，Ceph 对象网关将能够自动创建池。

在这种情况下，请确保在池中指定合理的 PG 数量。否则，网关使用默认编号，该编号很可能不适合您的需要。有关详细信息，请参阅每个池计算器的 Ceph Placement Group (PG)。

6. 在对象网关节点上，创建 **done** 文件：

```
# touch /var/lib/ceph/radosgw/ceph-rgw.`hostname -s`/done
```

7. 在对象网关节点上，更改所有者和组权限：

```
# chown -R ceph:ceph /var/lib/ceph/radosgw
# chown -R ceph:ceph /var/log/ceph
# chown -R ceph:ceph /var/run/ceph
# chown -R ceph:ceph /etc/ceph
```

8. 在 **Object Gateway** 节点上打开 **TCP** 端口 **8080**：

```
# firewall-cmd --zone=public --add-port=8080/tcp
# firewall-cmd --zone=public --add-port=8080/tcp --permanent
```

9. 在对象网关节点上，启动并启用 **ceph-radosgw** 进程：

语法

```
# systemctl enable ceph-radosgw.target
# systemctl enable ceph-radosgw@rgw.<rgw_hostname>
# systemctl start ceph-radosgw@rgw.<rgw_hostname>
```

示例

```
# systemctl enable ceph-radosgw.target
# systemctl enable ceph-radosgw@rgw.node1
# systemctl start ceph-radosgw@rgw.node1
```

安装后，如果在 **monitor** 上设置了写入功能，**Ceph** 对象网关会自动创建池。如需有关手动创建池的详细信息，请参阅存储策略指南中的池章节。

其它资源

- **Red Hat Ceph Storage 4** *对象网关配置和管理指南*

附录 C. 配置 ANSIBLE 清单位置

作为选项，您可以为 ceph-ansible 临时环境和生产环境配置清单位置文件。

先决条件

- Ansible 管理节点.
- 对 Ansible 管理节点的根级别访问权限.
- ceph-ansible 软件包安装在节点上。

流程

1. 进入 /usr/share/ceph-ansible 目录：

```
[ansible@admin ~]# cd /usr/share/ceph-ansible
```

2. 为临时和生产环境创建子目录：

```
[ansible@admin ceph-ansible]$ mkdir -p inventory/staging inventory/production
```

3. 编辑 ansible.cfg 文件并添加以下几行：

```
[defaults]
inventory = ./inventory/staging # Assign a default inventory directory
```

4. 为每个环境创建一个清单"主机"文件：

```
[ansible@admin ceph-ansible]$ touch inventory/staging/hosts
[ansible@admin ceph-ansible]$ touch inventory/production/hosts
```

   a. 打开并编辑每个 hosts 文件，并在 [mons] 部分下添加 Ceph 监控节点：

```
[mons]
MONITOR_NODE_NAME_1
```

```
MONITOR_NODE_NAME_1
MONITOR_NODE_NAME_1
```

示例

```
[mons]
mon-stage-node1
mon-stage-node2
mon-stage-node3
```

注意

默认情况下，playbook 在暂存环境中运行。在生产环境中运行 playbook:

```
[ansible@admin ceph-ansible]$ ansible-playbook -i inventory/production
playbook.yml
```

其它资源

- **有关安装 ceph-ansible 软件包的更多信息，请参阅安装红帽存储集群。**

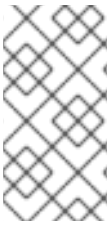
附录 D. 覆盖 CEPH 默认设置

除非在 Ansible 配置文件中另有指定，否则 Ceph 将使用其默认设置。

由于 Ansible 管理 Ceph 配置文件，请编辑 /usr/share/ceph-ansible/group_vars/all.yml 文件，以更改 Ceph 配置。使用 ceph_conf_overrides 设置覆盖默认的 Ceph 配置。

Ansible 支持与 Ceph 配置文件相同的部分； [global]、[mon]、[osd]、[mds]、[rgw] 等。您还可以覆盖特定的实例，如特定的 Ceph 对象网关实例。例如：

```
###################
# CONFIG OVERRIDE #
###################

ceph_conf_overrides:
  client.rgw.server601.rgw1:
    rgw_enable_ops_log: true
    log_file: /var/log/ceph/ceph-rgw-rgw1.log
```

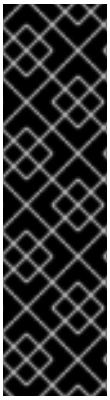


注意

不要将变量用作 ceph_conf_overrides 设置中的键。您必须为主机传递要覆盖特定配置值的部分的绝对标签。

注意

当引用 Ceph 配置文件的特定部分时，Ansible 不包含大括号。部分和设置名称以冒号结尾。



重要

不要使用 CONFIG OVERRIDE 部分中的 cluster_network 参数设置集群网络，因为这可能导致 Ceph 配置文件中设置两个相互冲突的集群网络。

要设置集群网络，请使用 CEPH CONFIGURATION 部分中的 cluster_network 参数。详情请参阅 *Red Hat Ceph Storage 指南*中的*安装 Red Hat Ceph Storage 集群*。

附录 E. 将现有 CEPH 集群导入到 ANSIBLE

您可以将 Ansible 配置为使用在没有 Ansible 的情况下部署的集群。例如，如果您将 Red Hat Ceph Storage 1.3 集群升级到版本 2，请按照以下步骤将其配置为使用 Ansible：

1. 从 1.3 手动升级到版本 2 后，在管理节点上安装和配置 Ansible。

2. 确保 Ansible 管理节点对集群中的所有 Ceph 节点进行免密码 ssh 访问。详情请查看 第 3.9 节 “为 Ansible 启用免密码 SSH”。

3. 以 root 用户身份，在 /etc/ansible/ 目录中创建一个指向 Ansible group_vars 目录的符号链接：

```
# ln -s /usr/share/ceph-ansible/group_vars /etc/ansible/group_vars
```

4. 以 root 用户身份，使用 all.yml.sample 文件中创建一个 all.yml 文件，并打开该文件进行编辑：

```
# cd /etc/ansible/group_vars
# cp all.yml.sample all.yml
# vim all.yml
```

5. 在 group_vars/all.yml 中，将 generate_fsid 设置为 false。

6. 通过执行 ceph fsid 获得当前集群 fsid。

7. 在 group_vars/all.yml 中设置检索到的 fsid。

8. 修改 /etc/ansible/hosts 中的 Ansible 清单，使其包含 Ceph 主机。在 [mons] 部分下添加监视器，在 [osds] 部分下的 OSD 和网关下的 [rgws] 部分下将其角色标识到 Ansible。

9. 确定 ceph_conf_overrides 已更新，使用用于 all.yml 文件中的 [global]、[osd]、[mon] 和 [client] 项的原始 ceph.conf 选项。

   在 ceph_conf_overrides 中不应添加 osd journal、public_network 和 cluster_network 等

选项，因为它们已经是 all.yml 的一部分。仅应将不属于 all.yml 且位于原始 ceph.conf 中的选项添加到 ceph_conf_overrides。

10. **从 /usr/share/ceph-ansible/ 目录运行 playbook。**

```
# cd /usr/share/ceph-ansible/
# ansible-playbook infrastructure-playbooks/take-over-existing-cluster.yml -u <username> -i
hosts
```

附录 F. 清除 ANSIBLE 部署的存储集群

如果您不再使用 Ceph 存储集群，则使用 purge-docker-cluster.yml playbook 来删除集群。当安装过程失败且您要重新开始时，清除存储集群也很有用。

警告

在清除 Ceph 存储集群后，OSD 上的所有数据都会永久丢失。

先决条件

- 对 Ansible 管理节点的根级别访问权限.
- 访问 ansible 用户帐户.
- 对于裸机部署：
  - 如果 /usr/share/ceph-ansible/group-vars/osds.yml 文件中的 osd_auto_discovery 选项设为 true，则 Ansible 将无法清除存储集群。因此，注释掉 osd_auto_discovery，并在 osds.yml 文件中声明 OSD 设备。
- 确保 /var/log/ansible/ansible.log 文件可由 ansible 用户帐户写入。

流程

1. 进入 /usr/share/ceph-ansible/ 目录：

```
[root@admin ~]# cd /usr/share/ceph-ansible
```

2. 以 ansible 用户身份，运行清除 playbook。

   a. 对于 裸机 部署，请使用 purge-cluster.yml playbook 来清除 Ceph 存储集群：

```
[ansible@admin ceph-ansible]$ ansible-playbook infrastructure-playbooks/purge-cluster.yml
```

b. **对于容器部署：**

i. **使用 purge-docker-cluster.yml playbook 来清除 Ceph 存储集群：**

```
[ansible@admin ceph-ansible]$ ansible-playbook infrastructure-playbooks/purge-docker-cluster.yml
```

注意

此 playbook 删除 Ceph Ansible playbook 创建的所有软件包、容器、配置文件和所有数据。

ii. **要指定非默认清单文件 (/etc/ansible/hosts)，请使用 -i 参数：**

语法

```
[ansible@admin ceph-ansible]$ ansible-playbook infrastructure-playbooks/purge-docker-cluster.yml -i INVENTORY_FILE
```

替换

***INVENTORY_FILE*，使用带有清单文件的路径。**

示例

```
[ansible@admin ceph-ansible]$ ansible-playbook infrastructure-playbooks/purge-docker-cluster.yml -i ~/ansible/hosts
```

iii. **要跳过移除 Ceph 容器镜像，请使用 --skip-tags=”remove_img” 选项：**

```
[ansible@admin ceph-ansible]$ ansible-playbook --skip-tags="remove_img"
infrastructure-playbooks/purge-docker-cluster.yml
```

iv. **要跳过删除在安装过程中安装的软件包的过程，请使用 --skip-tags="with_pkg" 选项：**

```
[ansible@admin ceph-ansible]$ ansible-playbook --skip-tags="with_pkg"
infrastructure-playbooks/purge-docker-cluster.yml
```

其它资源

- **如需了解更多详细信息，请参阅 *OSD Ansible 设置*。**

附录 G. 使用 ANSIBLE 清除 CEPH 仪表板

如果不再需要安装仪表板，请使用 purge-dashboard.yml playbook 删除仪表板。在对仪表板或其组件进行故障排除时，您可能还要清除仪表板。

先决条件

- Red Hat Ceph Storage 4.3 或更高版本。
- ceph-ansible 附带最新版本的 Red Hat Ceph Storage。
- 对存储集群中所有节点的 sudo 级别访问权限。

流程

1. 登录 Ansible 管理节点。
2. 进入 /usr/share/ceph-ansible/ 目录：

   示例

   ```
   [ansible@admin ~]$ cd /usr/share/ceph-ansible/
   ```

3. 运行 Ansible purge-dashboard.yml playbook，并在提示时输入 yes 来确认仪表板的清除：

   示例

   ```
   [ansible@admin ceph-ansible]$ ansible-playbook infrastructure-playbooks/purge-dashboard.yml -i hosts -vvvv
   ```

验证

- 运行 **ceph mgr services** 命令以验证仪表板不再运行：

  语法

```
ceph mgr services
```

  控制面板 URL 不显示出来。

其它资源

- 要安装仪表板，请参阅在 ***Red Hat Ceph Storage Dashboard Guide*** 中使用 **Ansbile** *安装* 仪表板。

附录 H. 使用 ANSIBLE-VAULT 加密 ANSIBLE 密码变量

您可以使用 ansible-vault 来加密用于存储密码的 Ansible 变量，使它们不可读为纯文本。例如，在 group_vars/all.yml 中，ceph_docker_registry_username 和 ceph_docker_registry_password 变量可以设置为服务帐户凭证或客户门户凭证。该服务帐户经过设计为共享，但客户门户网站密码应该安全。除了加密 ceph_docker_registry_password 外，您可能还希望加密 dashboard_admin_password 和 grafana_admin_password。

先决条件

- 一个正在运行的 Red Hat Ceph Storage 集群。
- 访问 Ansible 管理节点.

流程

1. 登录 Ansible 管理节点。
2. 进入 /usr/share/ceph-ansible/ 目录：

```
[admin@admin ~]$ cd /usr/share/ceph-ansible/
```

3. 运行 ansible-vault 并创建新的 vault 密码：

示例

```
[admin@admin ceph-ansible]$ ansible-vault encrypt_string --stdin-name
'ceph_docker_registry_password_vault'
New Vault password:
```

4. 重新输入 vault 密码以确认它：

示例

```
[admin@admin ceph-ansible]$ ansible-vault encrypt_string --stdin-name
'ceph_docker_registry_password_vault'
New Vault password:
Confirm New Vault password:
```

5. **输入要加密的密码，然后输入 CTRL+D 两次以完成该条目：**

语法

```
ansible-vault encrypt_string --stdin-name 'ceph_docker_registry_password_vault'
New Vault password:
Confirm New Vault password:
Reading plaintext input from stdin. (ctrl-d to end input)
PASSWORD
```

使用密码替换 *PASSWORD*：

示例

```
[admin@admin ceph-ansible]$ ansible-vault encrypt_string --stdin-name
'ceph_docker_registry_password_vault'
New Vault password:
Confirm New Vault password:
Reading plaintext input from stdin. (ctrl-d to end input)
SecurePassword
```

在键入密码后请勿按 Enter 键，否则它将在加密字符串中包含新行作为密码的一部分。

6.

记录以 **ceph_docker_registry_password_vault: !vault |** 开头并以几行数字结束的输出，它将在下一步中使用：

示例

```
[admin@admin ceph-ansible]$ ansible-vault encrypt_string --stdin-name
'ceph_docker_registry_password_vault'
New Vault password:
Confirm New Vault password:
Reading plaintext input from stdin. (ctrl-d to end input)
SecurePasswordceph_docker_registry_password_vault: !vault |
          $ANSIBLE_VAULT;1.1;AES256

383836396461666561303266666332626438363439303738363763313264373530323761653
06234

316138633461663265353038323131663163646236376166 0a373338373346634343638653
56633

663839633230333036623337653839383536306234333465653635346364346436343364306
43438

613430666264636537 0a343135316633303830653565633736303466636261326361333766 6
13462
          393533653431373231633439376364646635343832343265316661393765616635 32
Encryption successful
```

您需要的输出在密码后马上开始，没有空格或新行。

7. **打开以编辑 group_vars/all.yml，并将上方的输出粘贴到文件中：**

示例

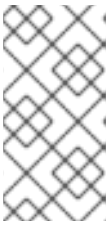
```
ceph_docker_registry_password_vault: !vault |
          $ANSIBLE_VAULT;1.1;AES256

383836396461666561303266666332626438363439303738363763313264373530323761653
06234

316138633461663265353038323131663163646236376166 0a373338373346634343638653
56633
```

```
6638396332303330366233376538393835363062343334656536353463643464364336430643438

6134306662646365370a343135316633303830653565633736303466636261326361333766613462
      393533653431373231633439376364646635343832343265316661393765616635 32
```

8. **在加密的密码下方添加一行：**

示例

```
ceph_docker_registry_password: "{{ ceph_docker_registry_password_vault }}"
```



注意

由于 Ansible 中的一个错误会在将 vault 值直接分配给 Ansible 变量时中断字符串类型，因此需要使用上述两个变量。

9. **配置 Ansible，以在运行 ansible-playbook 时要求输入 vault 密码。**

   a. **打开以编辑 /usr/share/ceph-ansible/ansible.cfg，并在 [defaults] 部分添加以下行：**

   ```
   ask_vault_pass = True
   ```

   b. **另外，您可以在每次运行 ansible-playbook 时传递 --ask-vault-pass ：**

   示例

   ```
   [admin@admin ceph-ansible]$ ansible-playbook -v site.yml --ask-vault-pass
   ```

10. **重新运行 site.yml 或 site-container.yml，以确保没有与加密密码相关的错误。**

示例

```
[admin@admin ceph-ansible]$ ansible-playbook -v site.yml -i hosts --ask-vault-pass
```

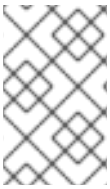
只有在您不使用 /etc/ansible/hosts 的默认 Ansible 清单位置时，才需要 -i hosts 选项。

其它资源

- **请参阅 Red Hat Container Registry 身份验证中的服务帐户信息**

附录 I. 常规 ANSIBLE 设置

以下是最常见的可配置 Ansible 参数。根据部署方法（裸机或容器），有两组参数。



注意

这不是所有可用 Ansible 参数的完整列表。

裸机和容器 设置

monitor_interface

Ceph 监控节点侦听的接口。

值

用户定义的

必填

是

备注

至少为一个 monitor_* 参数分配一个值。

monitor_address

Ceph 监控节点侦听的地址。

值

用户定义的

必填

是

备注

至少为一个 monitor_* 参数分配一个值。

monitor_address_block

Ceph 公共网络的子网。

值

用户定义的

必填

是

备注

当节点的 IP 地址未知但已知子网时，请使用。至少为一个 monitor_* 参数分配一个值。

ip_version

值

ipv6

必填

是，如果使用 IPv6 地址。

public_network

Ceph 公共网络的 IP 地址和子网掩码，或者对应的 IPv6 地址（若使用 IPv6）。

值

用户定义的

必填

是

备注

如需更多信息，请参阅*验证 Red Hat Ceph Storage 的网络配置*。

cluster_network

Ceph 集群网络的 IP 地址和子网掩码，或者对应的 IPv6 地址（若使用 IPv6）。

值

用户定义的

必填

否

备注

如需更多信息，请参阅*验证 Red Hat Ceph Storage 的网络配置*。

configure_firewall

Ansible 将尝试配置适当的防火墙规则。

值

true 或 false

必填

否

特定裸机的设置

ceph_origin

值

repository 或 distro 或 local

必填

是

备注

repository 代表 Ceph 将通过一个新的仓库安装。distro 值意味着不会添加单独的存储库文件，您将获得 Linux 发行版本中包含的任何 Ceph 版本。local 值表示将从本地计算机复制 Ceph 二进制文件。

ceph_repository_type

值

cdn 或 iso

必填

是

ceph_rhcs_version

值

4

必填

是

ceph_rhcs_iso_path

ISO 镜像的完整路径。

值

用户定义的

必填

是，如果 ceph_repository_type 设为 iso。

特定容器的设置

ceph_docker_image

值

rhceph/rhceph-4-rhel8 或 cephimageinlocalreg，如果使用本地 Docker registry）

必填

是

ceph_docker_image_tag

值

rhceph/rhceph-4-rhel8 的 latest 版本或 customtag 在本地 registry 配置中提供。

必填

是

containerized_deployment

值

true

必填

是

ceph_docker_registry

值

registry.redhat.io 或 *LOCAL_FQDN_NODE_NAME*（如果使用本地 Docker registry）。

必填

是

附录 J. OSD ANSIBLE 设置

以下是最常见的可配置 OSD Ansible 参数。

osd_auto_discovery

自动查找用作 OSD 的空设备。

值

false

必填

否

备注

无法与 设备 一起使用。不能与 purge-docker-cluster.yml 或 purge-cluster.yml 一起使用。要使用这些 playbook，请注释掉 osd_auto_discovery 并使用 设备 声明 OSD 设备。

devices

存储 Ceph 数据的设备列表。

值

用户定义的

必填

是，如果指定设备列表。

备注

使用 osd_auto_discovery 设置时无法使用。使用 devices 选项时，ceph-volume lvm batch 模式将创建优化的 OSD 配置。

dmcrypt

加密 OSD。

值

true

必填

否

备注

默认值为 false。

lvm_volumes

FileStore 或 BlueStore 字典列表。

值

用户定义的

必填

是的，如果没有使用 devices 参数定义存储设备。

备注

每一字典必须包含 data、journal 和 data_vg 键。任何逻辑卷或卷组都必须是名称，而不是完整路径。data 和 journal 键可以是逻辑卷 (LV) 或分区，但不能将一个日志用于多个 data LV。data_vg 键必须是包含 data LV 的卷组。（可选） journal_vg 键可用于指定包含 journal LV 的卷组（如果适用）。

osds_per_device

每个设备要创建的 OSD 数量。

值

用户定义的

必填

否

备注

默认值为 1。

osd_objectstore

OSD 的 Ceph 对象存储类型。

值

bluestore 或 filestore

必填

否

备注

默认值为 bluestore。升级需要。