SUSE Enterprise Storage 7.1

部署指南

部署指南

SUSE Enterprise Storage 7.1

作者：Tomáš Ba#ant、Alexandra Settle、Liam Proven

出版日期：2025-03-20

https://documentation.suse.com

版权所有 © 2020–2025 SUSE LLC 和贡献者。保留所有权利。

除非另有说明，否则本文档根据知识共享署名-相同方式共享 4.0 国际 (Creative Commons Attribution-ShareAlike 4.0 International, CC-BY-SA 4.0) 获得许可：https://creativecommons.org/licenses/by-sa/4.0/legalcode。

有关 SUSE 商标，请参见 http://www.suse.com/company/legal/ 。所有第三方商标均是其各自所有者的财产。商标符号（®、™ 等）代表 SUSE 及其关联公司的商标。星号 (*) 代表第三方商标。

本指南力求涵盖所有细节，但这不能确保本指南准确无误。SUSE LLC 及其关联公司、作者和译者对于可能出现的错误或由此造成的后果皆不承担责任。

目录

关于本指南 ix

1 可用文档 ix

2 提供反馈 x

3 文档约定 xi

4 支持 xii

SUSE Enterprise Storage 支持声明 xiii • 技术预览 xiv

5 Ceph 贡献者 xiv

6 本指南中使用的命令和命令提示符 xiv

与 Salt 相关的命令 xv • 与 Ceph 相关的命令 xv • 一般的 Linux 命令 xvi • 附加信息 xvi

I 介绍 SUSE ENTERPRISE STORAGE (SES) 1

1 SES 和 Ceph 2

1.1 Ceph 特性 2

1.2 Ceph 核心组件 3

RADOS 3 • CRUSH 4 • Ceph 节点和守护进程 4

1.3 Ceph 存储结构 6

存储池 6 • 归置组 6 • 示例 7

1.4 BlueStore 8

1.5 附加信息 10

2 硬件要求和建议 11

2.1 网络概览 11

网络建议 12

2.2 多体系结构配置 14

2.3 硬件配置 14

最低集群配置 14 • 建议的生产集群配置 16 • 多路径配置 17

2.4 对象存储节点 18

最低要求 18 • 最小磁盘大小 19 • BlueStore 的 WAL 和 DB 设备的建议大小 19 • 用于 WAL/DB 分区的 SSD 20 • 磁盘的最大建议数量 20

2.5 Monitor 节点 20

2.6 对象网关节点 21

2.7 元数据服务器节点 21

2.8 管理节点 21

2.9 iSCSI 网关节点 22

2.10 SES 和其他 SUSE 产品 22

SUSE Manager 22

2.11 名称限制 22

2.12 共享一台服务器的 OSD 和 Monitor 22

3 管理节点 HA 设置 24

3.1 管理节点的 HA 集群概述 24

3.2 构建具有管理节点的 HA 集群 25

II 部署 CEPH 集群 27

4 简介和常见任务 28

4.1 阅读发行说明 28

5 安装和配置 SUSE Linux Enterprise Server 30

6 部署 Salt 31

7 使用 ceph-salt 部署引导集群 34

7.1 安装 ceph-salt 34

7.2 配置集群属性 34

使用 ceph-salt 外壳 35 • 添加 Salt 受控端 37 • 指定由 cephadm 管理的 Salt 受控端 38 • 指定管理节点 38 • 指定第一个 MON/MGR 节点 39 • 指定调整的配置 39 • 生成 SSH 密钥对 40 • 配置时间服务器 40 • 配置 Ceph Dashboard 登录身份凭证 41 • 使用容器注册表 42 • 启用数据传输中加密 (msgr2) 48 • 配置集群网络 50 • 确认集群配置 50 • 导出集群配置 52

7.3 更新节点并引导最小集群 52

7.4 查看最后的步骤 54

7.5 禁用不安全的客户端 55

8 使用 cephadm 部署其余核心服务 56

8.1 ceph orch 命令 56

显示 orchestrator 状态 56 • 列出设备、服务和守护进程 56

8.2 服务和归置规范 58

创建服务规范 58 • 创建归置规范 59 • 应用集群规范 60 • 导出运行中集群的规范 60

8.3 部署 Ceph 服务 61

部署 Ceph Monitor 和 Ceph Manager 61 • 部署 Ceph OSD 63 • 部署元数据服务器 63 • 部署对象网关 64 • 部署 iSCSI 网关 67 • 部署 NFS Ganesha 69 • 部署 `rbd-mirror` 70 • 部署监控堆栈 70

9 部署附加服务 73

9.1 安装 iSCSI 网关 73

iSCSI 块存储 73 • 关于 `ceph-iscsi` 的一般信息 75 • 部署考虑事项 76 • 安装和配置 77 • 使用 `tcmu-runner` 导出 RADOS 块设备映像 85

III 从旧版本升级 86

10 从 SUSE Enterprise Storage 6 升级到版本 7.1 87

10.1 升级前 87

需考虑的要点 88 • 备份集群配置和数据 89 • 确认先前升级的步骤 89 • 更新集群节点并确认集群健康状况 90 • 确认对软件源和容器映像的访问权限 91

10.2 升级 Salt 主控端 92

10.3 升级 MON、MGR 和 OSD 节点 94

10.4 升级网关节点 95

10.5 安装 `ceph-salt` 并应用集群配置 96

10.6 升级并采用监控堆栈 98

10.7 重新部署网关服务 100

升级对象网关 100 • 升级 NFS Ganesha 102 • 升级元数据服务器 106 • 升级 iSCSI 网关 107

10.8 升级后清理 108

11 从 SUSE Enterprise Storage 7 升级到版本 7.1 111

11.1 升级前 111

需考虑的要点 111 • 备份集群配置和数据 112 • 确认对软件源和容器映像的访问权限 112

11.2 将每个集群节点上的 SUSE Linux Enterprise Server 迁移到 SUSE Linux Enterprise Server 15 SP3 版本 113

11.3 升级每个集群节点上与 SUSE Enterprise Storage 相关的软件包 113

11.4 升级现有 Ceph 集群服务 113

A 基于上游“Pacific”小版本的 Ceph 维护更新 114

词汇表 115

关于本指南

本指南重点介绍如何部署基本 Ceph 集群以及如何部署其他服务。此外，还介绍了从先前的产品版本升级到 SUSE Enterprise Storage 7.1 的步骤。

SUSE Enterprise Storage 7.1 是 SUSE Linux Enterprise Server 15 SP3 的一个扩展。它融合了 Ceph (http://ceph.com/) 存储项目的功能与 SUSE 的企业工程和支持。SUSE Enterprise Storage 7.1 为 IT 组织提供了部署分布式存储体系结构的能力，该体系结构可支持使用市售硬件平台的许多用例。

1 可用文档

注意：联机文档和最新更新

我们的产品文档可从 https://documentation.suse.com 获取，您也可以在此处找到最新更新，以及浏览或下载各种格式的文档。最新的文档更新以英语版本提供。

此外，您安装的系统的 `/usr/share/doc/manual` 下会提供产品文档。该文档包含在名为 `ses-manual_LANG_CODE` 的 RPM 软件包中。如果系统上尚未安装该包，请进行安装，例如：

```
# zypper install ses-manual_en
```

针对本产品提供的文档如下：

部署指南 (https://documentation.suse.com/ses/html/ses-all/book-storage-deployment.html)

本指南重点介绍如何部署基本 Ceph 集群以及如何部署其他服务。此外，还介绍了从先前的产品版本升级到 SUSE Enterprise Storage 7.1 的步骤。

管理和操作指南 (https://documentation.suse.com/ses/html/ses-all/book-storage-admin.html)

本指南重点介绍在部署基本 Ceph 集群（第 2 天操作）之后，作为管理员需要处理的例行任务。此外，还介绍了访问 Ceph 集群中所存储数据的所有支持方法。

安全强化指南 (https://documentation.suse.com/ses/html/ses-all/book-storage-security.html)

本指南重点介绍如何确保集群的安全。

查错指南 (https://documentation.suse.com/ses/html/ses-all/book-storage-troubleshooting.html)

本指南将带您了解运行 SUSE Enterprise Storage 7.1 时的各种常见问题以及与 Ceph 或对象网关等相关组件有关的其他问题。

SUSE Enterprise Storage for Windows 指南 (https://documentation.suse.com/ses/html/ses-all/book-storage-windows.html)

本指南介绍如何使用 Windows 驱动程序集成、安装和配置 Microsoft Windows 环境和 SUSE Enterprise Storage。

2 提供反馈

欢迎您对此文档提供反馈和贡献。反馈渠道包括：

服务请求和支持

有关产品可用的服务和支持选项，请参见 http://www.suse.com/support/。

要创建服务请求，需在 SUSE Customer Center 注册一个 SUSE 订阅。请转到 https://scc.suse.com/support/requests 并登录，然后点击新建。

Bug 报告

在 https://bugzilla.suse.com/ 中报告文档问题。报告问题需要 Bugzilla 帐户。

要简化此过程，可以使用本文档 HTML 版本中的标题旁边的报告文档 Bug 链接。这样，就会在 Bugzilla 中预先选择正确的产品和类别，并添加当前章节的链接。然后，您便可以立即开始键入 Bug 报告。

贡献

要帮助改进本文档，请使用本文档 HTML 版本中的标题旁边的编辑源代码链接。这些链接会将您转到 GitHub 上的源代码，在其中可以创建拉取请求。参与贡献需要 GitHub 帐户。

有关本文档使用的文档环境的详细信息，请参见软件源的 README（网址：https://github.com/SUSE/doc-ses）。

邮件

您也可以将有关本文档中的错误以及相关反馈发送至：doc-team@suse.com。请在其中包含文档标题、产品版本和文档发布日期。此外，请包含相关的章节号和标题（或者提供URL），并提供问题的简要说明。

3 文档约定

本文档中使用了以下通知和排版约定：

- /etc/passwd：目录名称和文件名
- PLACEHOLDER：将会使用实际的值替换 PLACEHOLDER
- PATH：环境变量
- **ls**、--help：命令、选项和参数
- user：用户或组的名称
- package_name：软件包的名称
- Alt 、 Alt – F1 ：按键或组合键。按键以大写字母显示，与键盘上的一样。
- 文件、 文件 › 另存为：菜单项，按钮
- AMD/Intel 本段内容仅与 AMD64/Intel 64 体系结构相关。箭头标记文本块的开始位置和结束位置。
  IBM Z, POWER 本段内容仅与 IBM Z 和 POWER 体系结构相关。箭头标记文本块的开始位置和结束位置。
- 第 1 章“示例章节”：到本指南中另一章节的交叉引用。
- 必须使用 root 特权运行的命令。您往往还可以在这些命令前加上 **sudo** 命令，以非特权用户身份来运行它们。

```
# command
> sudo command
```

- 可以由非特权用户运行的命令。

```
> command
```

- 注意

警告：警报通知

在继续操作之前，您必须了解的不可或缺的信息。向您指出有关安全问题、潜在数据丢失、硬件损害或物理危害的警告。

重要：重要通知

在继续操作之前，您必须了解的重要信息。

注意：注意通知

额外信息，例如有关软件版本差异的信息。

提示：提示通知

有用信息，例如指导方针或实用性建议。

- 精简通知

额外信息，例如有关软件版本差异的信息。

有用信息，例如指导方针或实用性建议。

4 支持

下面提供了 SUSE Enterprise Storage 的支持声明和有关技术预览的一般信息。有关产品生命周期的细节，请参见 https://www.suse.com/lifecycle。

如果您有权获享支持，可在 https://documentation.suse.com/sles-15/html/SLES-all/cha-adm-support.html 中查找有关如何收集支持票据所需信息的细节。

4.1 SUSE Enterprise Storage 支持声明

要获得支持，您需要一个适当的 SUSE 订阅。要查看为您提供的具体支持服务，请转到 https://www.suse.com/support/ 并选择您的产品。

支持级别的定义如下：

L1

问题判定，该技术支持级别旨在提供兼容性信息、使用支持、持续维护、信息收集，以及使用可用文档进行基本查错。

L2

问题隔离，该技术支持级别旨在分析数据、重现客户问题、隔离问题领域，并针对级别 1 不能解决的问题提供解决方法，或作为级别 3 的准备级别。

L3

问题解决，该技术支持级别旨在借助工程方法解决级别 2 支持所确定的产品缺陷。

对于签约的客户与合作伙伴，SUSE Enterprise Storage 将为除以下软件包外的其他所有软件包提供 L3 支持：

- 技术预览。
- 声音、图形、字体和作品。
- 需要额外客户合同的软件包。
- 模块 Workstation Extension 随附的某些软件包仅享受 L2 支持。
- 名称以 `-devel` 结尾的软件包（包含头文件和类似的开发人员资源）只能与其主软件包一起获得支持。

SUSE 仅支持使用原始软件包，即，未发生更改且未重新编译的软件包。

4.2 技术预览

技术预览是 SUSE 提供的旨在让用户大致体验未来创新的各种软件包、堆栈或功能。随附这些技术预览只是为了提供方便，让您有机会在自己的环境中测试新的技术。非常希望您能提供反馈！如果您测试了技术预览，请联系 SUSE 代表，将您的体验和用例告知他们。您的反馈对于我们的未来开发非常有帮助。

技术预览存在以下限制：

- 技术预览仍处于开发阶段。因此，它们的功能可能不完备、不稳定，或者在其他方面不适合用于生产。
- 技术预览不受支持。
- 技术预览可能仅适用于特定的硬件体系结构。
- 技术预览的细节和功能可能随时会发生变化。因此，可能无法升级到技术预览的后续版本，而只能进行全新安装。
- SUSE 可能会发现某个预览不符合客户或市场需求，或者未遵循企业标准。可随时从产品中删除技术预览。SUSE 不承诺未来将提供此类技术的受支持版本。

有关产品随附的技术预览的概述，请参见 https://www.suse.com/releasenotes/x86_64/SUSE-Enterprise-Storage/7.1 上的发行说明。

5 Ceph 贡献者

Ceph 项目及其文档是数百个贡献者和组织辛勤工作的成果。有关详细信息，请参见 https://ceph.com/contributors/。

6 本指南中使用的命令和命令提示符

作为 Ceph 集群管理员，您需要通过运行特定命令来配置和调整集群行为。您将需要运行以下几种类型的命令：

6.1 与 Salt 相关的命令

这些命令可帮助您部署 Ceph 集群节点、同时在数个（或所有）集群节点上运行命令，或在您添加或删除集群节点时为您提供协助。最常用的命令是 **ceph-salt** 和 **ceph-salt config**。您需要以 root 身份在 Salt 主控端节点上运行 Salt 命令。通过以下提示符来引入这些命令：

```
root@master #
```

例如：

```
root@master # ceph-salt config ls
```

6.2 与 Ceph 相关的命令

这些是较低级别的命令，用于在命令行上配置和微调集群及其网关的所有方面，例如 **ceph**、**cephadm**、**rbd** 或 **radosgw-admin**。

要运行与 Ceph 相关的命令，您需要拥有 Ceph 密钥的读取访问权限，而密钥的用户权限则定义您在 Ceph 环境内的权限。一种方案是以 root 身份（或通过 **sudo**）运行 Ceph 命令，并使用不受限的默认密钥环“ceph.client.admin.key”。

建议您使用更安全的方案，即为每个管理员用户创建限制性更高的单独密钥，并将其存放在用户可读取的目录中，例如：

```
~/.ceph/ceph.client.USERNAME.keyring
```

提示：Ceph 密钥的路径

要使用自定义管理员用户和密钥环，每次运行 **ceph** 命令（使用 -n client.USER_NAME 和 --keyring PATH/TO/KEYRING 选项）时，都需要指定该密钥的用户名和路径。

为避免出现此情况，请将这些选项包含在各个用户的 ~/.bashrc 文件中的 CEPH_ARGS 变量中。

虽然您可以在任何集群节点上运行与 Ceph 相关的命令，但建议您在管理节点上运行这些命令。本文档使用 cephuser 用户来运行命令，因此通过以下提示符来引入命令：

```
cephuser@adm >
```

例如：

```
cephuser@adm > ceph auth list
```

提示：特定节点的命令

如果文档指示您在集群节点上以特定角色来运行命令，应通过该提示符来寻址。例如：

```
cephuser@mon >
```

6.2.1 运行 `ceph-volume`

从 SUSE Enterprise Storage 7 开始，Ceph 服务以容器化方式运行。如果您需要在 OSD 节点上运行 **`ceph-volume`**，则需要在其前面追加 **`cephadm`** 命令，例如：

```
cephuser@adm > cephadm ceph-volume simple scan
```

6.3 一般的 Linux 命令

与 Ceph 无关的 Linux 命令（例如 **`mount`**、**`cat`** 或 **`openssl`**）可通过 `cephuser@adm >` 或 `#` 提示符来引入，具体取决于相关命令所需的特权。

6.4 附加信息

有关 Ceph 密钥管理的详细信息，请参见《管理和操作指南》, 第 30 章 “cephx 身份验证”，第 30.2 节 “主要管理”。

I 介绍 SUSE Enterprise Storage (SES)

1 SES 和 Ceph 2

2 硬件要求和建议 11

3 管理节点 HA 设置 24

1 SES 和 Ceph

SUSE Enterprise Storage 是基于 Ceph 技术的分布式存储系统，旨在提高可伸缩性、可靠性和性能。Ceph 集群可在常见网络（例如以太网）中的市售服务器上运行。该集群能够正常扩展到数千台服务器（后面称为“节点”）以及 PB 量级的处理能力。与使用分配表来存储和提取数据的传统系统相反，Ceph 使用确定性算法来为数据分配存储空间，并且不采用任何集中式信息结构。Ceph 假设在存储集群中，硬件的添加或删除属于惯例，而不是例外情况。Ceph 集群可将数据分布和重新分布、数据复制、故障检测和恢复等管理任务自动化。Ceph 既可自我修复，又可自我管理，因此可以降低管理开销和预算开销。

本章提供 SUSE Enterprise Storage 7.1 的综合概述，并简要介绍一些最重要的组件。

1.1 Ceph 特性

Ceph 环境具有以下特性：

可伸缩性

Ceph 可扩展到数千个节点，并可管理 PB 量级的存储。

市售硬件

无需特殊的硬件即可运行 Ceph 集群。有关详细信息，请参见第 2 章 “硬件要求和建议”

自我管理

Ceph 集群可自我管理。添加、删除节点或节点发生故障时，集群可自动重新分布数据。此外，它还能识别过载的磁盘。

无单一故障点

重要的信息不会单独存储在集群中的某个节点上。可以配置冗余数量。

开放源代码软件

Ceph 是一套开源软件解决方案，独立于特定的硬件或供应商。

1.2 Ceph 核心组件

要充分利用 Ceph 的强大功能，需要了解一些基本的组件和概念。本节介绍经常在其他章节中提到的 Ceph 的某些组成部分。

1.2.1 RADOS

Ceph 的基本组件称为 **RADOS （可靠自主分布式对象存储）**。该组件负责管理集群中存储的数据。Ceph 中的数据通常以对象的形式存储。每个对象由标识符和数据组成。

RADOS 提供以下方法来访问涉及许多用例的存储对象：

对象网关

对象网关是 RADOS 对象存储区的 HTTP REST 网关。使用该网关可以直接访问 Ceph 集群中存储的对象。

RADOS 块设备

可以像访问任何其他块设备一样访问 RADOS 块设备 (RBD)。例如，可将这些设备与 `libvirt` 结合使用，以实现虚拟化。

CephFS

Ceph 文件系统是符合 POSIX 标准的文件系统。

`librados`

`librados` 是一个库，可与许多编程语言结合使用，以创建能够直接与存储集群交互的应用。

`librados` 由对象网关和 RBD 使用，而 CephFS 直接与 RADOS 连接。请参见（图 1.1 “与 Ceph 对象存储连接”）。

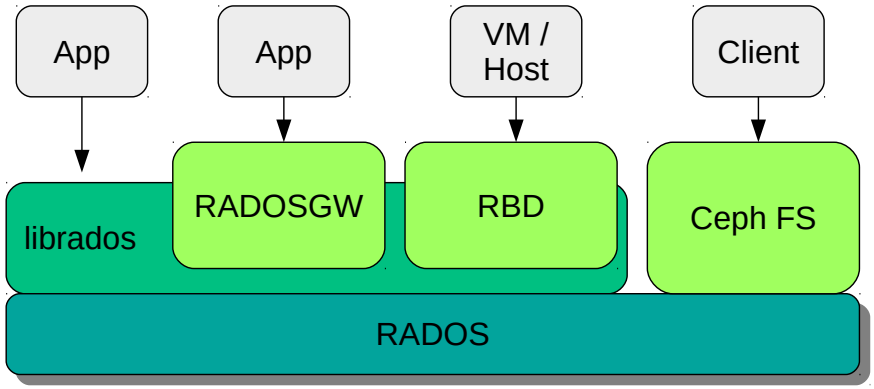

图 1.1： 与 CEPH 对象存储连接

1.2.2 CRUSH

Ceph 集群的核心是 **CRUSH** 算法。CRUSH 是 **Controlled Replication Under Scalable Hashing**（基于可缩放哈希的受控复制）的缩写。CRUSH 是处理存储分配的函数，所需的参数相对较少。这意味着，只需提供少量的信息就能计算对象的存储位置。参数是集群的现行对照，包括健康状况、管理员定义的某些放置规则，以及需要存储或检索的对象名称。提供这些信息后，Ceph 集群中的所有节点即可计算对象及其副本的存储位置。这使数据写入或读取变得非常有效。CRUSH 会尝试在集群中的所有节点上均匀分布数据。

CRUSH 索引包含所有存储节点，以及管理员定义的有关在集群中存储对象的归置规则。该索引定义了通常对应于集群物理结构的分层结构。例如，包含数据的磁盘位于主机中，主机位于机柜中，机柜位于设备排中，而设备排位于数据中心中。此结构可用于定义**故障域**。然后，Ceph 可确保将复制项存储在特定故障域的不同分支中。

如果将故障域设置为机柜，则在不同的机柜中分布对象复制项。这样就可以缓解机柜中的交换机发生故障所造成的服务中断。如果某个电源分配单元为一排机柜供电，则可将故障域设置为设备排。当电源分配单元发生故障时，其他设备排中仍可提供复制的数据。

1.2.3 Ceph 节点和守护进程

在 Ceph 中，节点是为集群工作的服务器。它们可以运行多种不同类型的守护进程。我们建议在每个节点上只运行一种类型的守护进程，但 Ceph Manager 守护进程除外，此类守护进程可与 Ceph Monitor 共置。每个集群都至少需要运行 Ceph Monitor、Ceph Manager 和 OSD 守护进程：

管理节点

管理节点是您在其中运行相应命令来管理集群的一种 Ceph 集群节点。管理节点是 Ceph 集群的中心点，因为它管理着集群的其余节点，它会查询这些节点的 Salt 受控端服务并向其发出指示。

Ceph Monitor

Ceph Monitor（往往缩写为 **MON**）节点负责维护有关集群健康状况、所有节点的索引和数据分布规则的信息（请参见第 1.2.2 节 “CRUSH”）。

如果发生故障或冲突，集群中的 Ceph Monitor 节点会根据多数派原则确定哪些信息是正确的。为了构成有效的多数派，建议设置奇数数量的 Ceph Monitor 节点，并至少设置三个这样的节点。

如果使用多个站点，应在奇数个站点之间分布 Ceph Monitor 节点。每个站点的 Ceph Monitor 节点数应满足以下要求：当一个站点发生故障时，50% 以上的 Ceph Monitor 节点可保持正常运行。

Ceph Manager

Ceph Manager 会从整个集群收集状态信息。Ceph Manager 守护进程与 Ceph Monitor 守护进程一同运行。它提供附加的监视功能，并与外部监视和管理系统连接。它还包含其他服务。例如，Ceph Dashboard Web UI 会在 Ceph Manager 所在的节点上运行。

Ceph Manager 不需要额外的配置，只需确保它在运行即可。

Ceph OSD

Ceph OSD 是一个守护进程，负责处理属于物理或逻辑存储单元（硬盘或分区）的**对象存储设备**。对象存储设备可以是物理磁盘/分区，也可以是逻辑卷。此外，该守护进程会处理数据复制，并在添加或删除节点后进行重新平衡。

Ceph OSD 守护进程与 Monitor 守护进程通讯，并为其提供其他 OSD 守护进程的状态。

要使用 CephFS、对象网关、NFS Ganesha 或 iSCSI 网关，还需要其他节点：

元数据服务器 (MDS)

CephFS 元数据存储在自己的 RADOS 存储池中（请参见第 1.3.1 节 “存储池”）。元数据服务器充当着元数据的智能缓存层，会根据需要对访问进行序列化。这样，无需显式同步，许多客户端便可实现并发访问。

对象网关

对象网关是 RADOS 对象存储的 HTTP REST 网关。它与 OpenStack Swift 和 Amazon S3 兼容，具有自己的用户管理功能。

NFS Ganesha

使用 NFS Ganesha 可通过 NFS 访问对象网关或 CephFS。该组件在用户空间而不是内核空间中运行，直接与对象网关或 CephFS 交互。

iSCSI 网关

iSCSI 是一种存储网络协议，可让客户端将 SCSI 命令发送到远程服务器上的 SCSI 存储设备（目标）。

Samba 网关

Samba 网关提供对 CephFS 上所存储数据的 Samba 访问途径。

1.3 Ceph 存储结构

1.3.1 存储池

Ceph 集群中存储的对象放置在**存储池**中。对外部环境而言，存储池代表集群的逻辑分区。对于每个存储池，可以定义一组规则，例如，每个对象必须有多少个复制项。存储池的标准配置称为**副本存储池**。

存储池通常包含多个对象，但也可以将其配置为充当类似 RAID 5 的作用。在此配置中，对象连同其他编码块一起存储在块中。编码块包含冗余信息。管理员可以定义数据块和编码块的数量。在此配置中，存储池称为**纠删码存储池**或 **EC 存储池**。

1.3.2 归置组

归置组 (PG) 用于在存储池中分布数据。创建存储池时，会设置特定数目的归置组。归置组在内部用于将对象分组，是影响 Ceph 集群性能的重要因素。对象的 PG 根据该对象的名称确定。

1.3.3 示例

本节提供有关 Ceph 如何管理数据的简化示例（请参见图 1.2 “小规模 Ceph 示例”）。此示例并不代表 Ceph 集群的建议配置。硬件设置由三个存储节点或 Ceph OSD（主机 1、主机 2 和主机 3）组成。每个节点包含三个用作 OSD（osd.1 到 osd.9）的硬盘。此示例中忽略了 Ceph Monitor 节点。

注意：Ceph OSD 与 OSD 之间的差别

尽管 **Ceph OSD** 或 **Ceph OSD 守护进程**是指节点上运行的守护进程，但 **OSD** 一词指的是与守护进程交互的逻辑磁盘。

集群包含两个存储池：存储池 A 和存储池 B。尽管存储池 A 仅复制对象两次，但存储池 B 的恢复能力更重要，该存储池中的每个对象都有三个复制项。

当应用将某个对象放入存储池中时（例如，通过 REST API），将会根据存储池和对象名称选择归置组（PG1 到 PG4）。然后，CRUSH 算法会根据包含对象的归置组，计算要将对象存储到哪些 OSD。

在此示例中，故障域设置为主机。这可确保将对象的复制项存储在不同的主机上。根据针对存储池设置的复制级别，对象将存储在归置组使用的两个或三个 OSD 上。

写入对象的应用只与一个 Ceph OSD（主要 Ceph OSD）交互。主要 Ceph OSD 处理复制过程，并在所有其他 OSD 存储对象后确认写入过程完成。

如果 osd.5 发生故障，osd.1 上仍会提供 PG1 中的所有对象。一旦集群发现某个 OSD 发生故障，另一个 OSD 会立即接管工作。在此示例中，osd.4 用于取代 osd.5。然后，osd.1 上存储的对象会复制到 osd.4，以恢复复制级别。

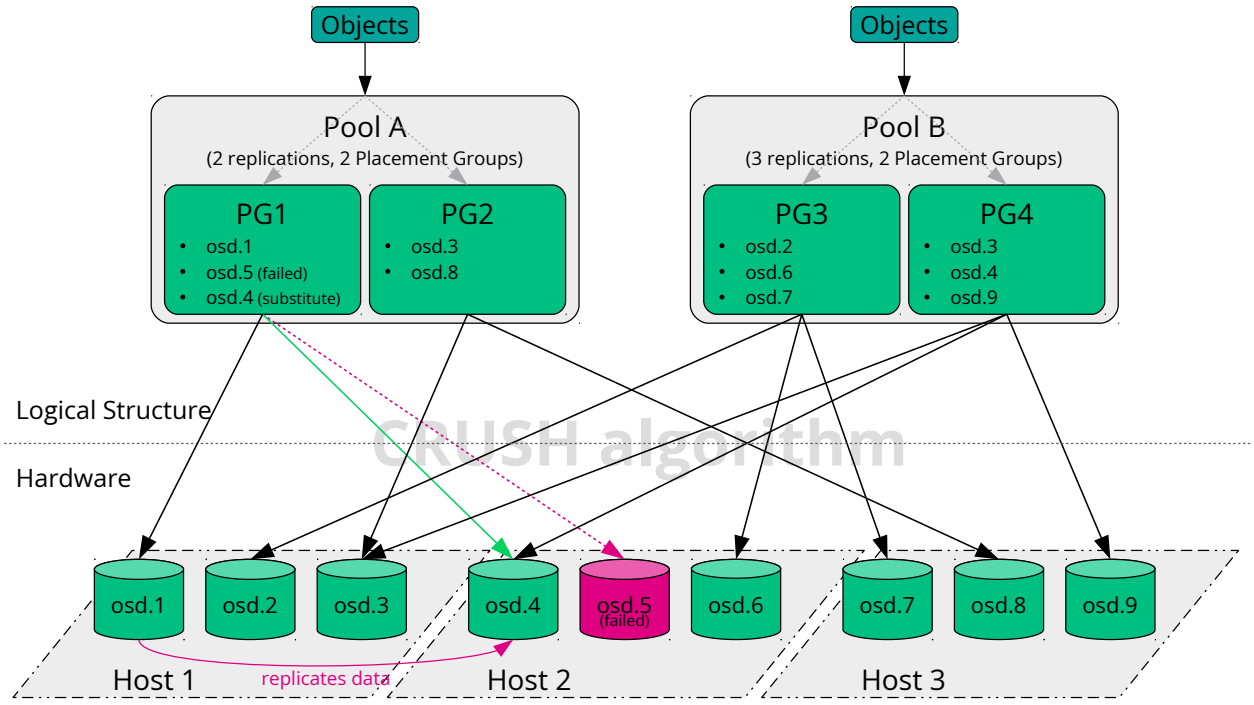

图 1.2： 小规模 CEPH 示例

如果将包含新 OSD 的新节点添加到集群，集群索引将会更改。然后，CRUSH 函数将返回对象的不同位置。接收新位置的对象将被重新定位。此过程将使得所有 OSD 被平衡使用。

1.4 BlueStore

从 SES 5 开始，使用 BlueStore 作为 Ceph 的新默认存储后端。BlueStore 的性能优于 FileStore，具有全数据校验和及内置压缩功能。

它可管理一至三个存储设备。在最简单的情况下，BlueStore 会使用一个主存储设备。该设备通常被分割成两个分区：

1. 一个名为 BlueFS 的较小分区，用于实施 RocksDB 所需的类似文件系统的功能。
2. 其余部分通常是一个较大的分区，占用了设备的剩余空间。它直接由 BlueStore 管理，包含所有实际数据。此主设备通常通过数据目录中的块符号链接来标识。

还可跨两个额外的设备来部署 BlueStore：

WAL 设备可用于存储 BlueStore 的内部日志或预写日志。它通过数据目录中的 `block.wal` 符号链接来标识。只有 WAL 设备速度比主设备或 DB 设备快时，使用单独的 WAL 设备才比较实用，例如，下列情况就很适合：

- WAL 设备是 NVMe，DB 设备是 SSD，数据设备是 SSD 或 HDD。
- WAL 和 DB 设备都是单独的 SSD，数据设备是 SSD 或 HDD。

DB 设备可用于存储 BlueStore 的内部元数据。BlueStore（更确切地说，是嵌入式 RocksDB）会将尽可能多的元数据存放于 DB 设备，以提升性能。再次重申，只有共享 DB 设备速度比主设备快时，供应共享 DB 设备才有所助益。

提示：规划 DB 大小

规划时请考虑充分，以确保为 DB 设备分配足够的大小。如果 DB 设备填满，元数据将溢出到主设备，这会严重降低 OSD 的性能。

您可以使用 **`ceph daemon osd`**`.ID` **`perf dump`** 命令来检查 WAL/DB 分区是否即将填满及溢出。`slow_used_bytes` 值显示即将溢出的数据量：

```
cephuser@adm > ceph daemon osd.ID perf dump | jq '.bluefs'
"db_total_bytes": 1073741824,
"db_used_bytes": 33554432,
"wal_total_bytes": 0,
"wal_used_bytes": 0,
"slow_total_bytes": 554432,
"slow_used_bytes": 554432,
```

1.5 附加信息

- 作为一个社区项目，Ceph 自身具有丰富的联机文档。对于本手册中未介绍的主题，请参见 https://docs.ceph.com/en/pacific/。
- 由 **S.A. Weil、S.A. Brandt、E.L. Miller 和 C. Maltzahn** 撰写的原始文献 **CRUSH: Controlled, Scalable, Decentralized Placement of Replicated Data**（CRUSH：复制数据的受控、可缩放、分布式放置）提供了有关 Ceph 内部工作原理的有用见解。特别是在部署大规模集群时，推荐您阅读此文章。可在 http://www.ssrc.ucsc.edu/papers/weil-sc06.pdf 上找到该文献。
- SUSE Enterprise Storage 可与非 SUSE OpenStack 发行套件搭配使用。Ceph 客户端需与 SUSE Enterprise Storage 兼容。

注意

SUSE 支持 Ceph 部署的服务器组件，而客户端则由 OpenStack 发行套件供应商提供支持。

2 硬件要求和建议

Ceph 的硬件要求在很大程度上取决于 IO 工作负载。在着手进行详细规划时，应考虑以下硬件要求和建议。

一般情况下，本节所述的建议是按进程提出的。如果同一台计算机上有多个进程，则需要提高 CPU、RAM、磁盘和网络要求。

2.1 网络概览

Ceph 有以下几个逻辑网络：

- 名为公共网络的前端网络。
- 名为集群网络的可信内部网络（后端网络）。此项是可选的。
- 网关的一个或多个客户端网络。此为可选网络，不在本章的讨论范围之内。

公共网络是 Ceph 守护进程相互之间以及与其客户端之间进行通讯的网络。这意味着所有 Ceph 集群流量都通过该网络传输（配置集群网络时除外）。

集群网络是 OSD 节点之间的后端网络，用于执行复制、重新平衡和恢复操作。如果配置了此可选网络，则理想情况下，它将提供两倍于公共网络的带宽（默认为三向复制），因为主 OSD 会通过此网络向其他 OSD 发送两个副本。公共网络位于客户端和网关之间，用于在其中一端与 Monitor、Manager、MDS 节点、OSD 节点进行通讯。Monitor、Manager 和 MDS 节点也会使用该网络与 OSD 节点进行通讯。

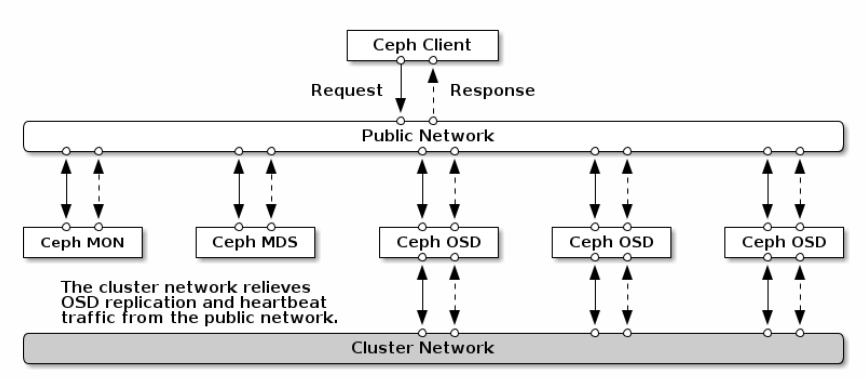

图 2.1： 网络概览

2.1.1 网络建议

我们建议使用具有足够带宽的单一容错网络，以满足您的需求。对于 Ceph 公共网络环境，我们建议使用通过 802.3ad (LACP) 绑定的两个绑定 25 GbE（或更快的）网络接口。这被视为 Ceph 的最小设置。如果您还使用集群网络，则建议您使用四个绑定的 25 GbE 网络接口。绑定两个或更多网络接口可通过链接聚合提供更高的吞吐量，并可提供冗余链接和交换机，进而提高了容错性和可维护性。

您还可以创建 VLAN 来隔离通过绑定传输的不同类型的流量。例如，您可以创建一个绑定来提供两个 VLAN 接口，一个用于公共网络，另一个用于集群网络。但在设置 Ceph 网络时**不**需要这样做。有关绑定接口的详细信息，请参见 https://documentation.suse.com/sles/15-SP3/html/SLES-all/cha-network.html#sec-network-iface-bonding 。

通过将组件隔离到故障域中，可以提高容错性。为了提高网络的容错性，可从两个独立的网络接口卡 (NIC) 绑定一个接口，以在单个 NIC 出现故障时提供保护。同样，在两个交换机之间创建一个绑定可在单个交换机出现故障时提供保护。建议您与网络设备供应商协商，以构造所需的容错级别。

重要：不支持管理网络

其他管理网络设置（例如可实现 SSH、Salt 或 DNS 网络分隔的设置）未经过测试也不受支持。

提示：通过 DHCP 配置的节点

如果存储节点是通过 DHCP 配置的，则默认超时可能会不够长，无法保证在各个 Ceph 守护进程启动前正确配置网络。如果发生这种情况，Ceph MON 和 OSD 将无法正确启动（运行 `systemctl status ceph\*` 将产生“无法绑定”错误）。为避免发生此问题，建议将您存储集群中每个节点上的 DHCP 客户端超时至少增加到 30 秒。为此，可在每个节点上更改以下设置：

在 `/etc/sysconfig/network/dhcp` 中，设置

```
DHCLIENT_WAIT_AT_BOOT="30"
```

在 `/etc/sysconfig/network/config` 中，设置

```
WAIT_FOR_INTERFACES="60"
```

2.1.1.1 将专用网络添加到正在运行的集群

如果您在部署 Ceph 期间未指定集群网络，则系统假设使用的是单个公共网络环境。尽管 Ceph 可在公共网络中正常运行，但如果您设置了另一个专用集群网络，Ceph 的性能和安全性将会得到提升。要支持两个网络，每个 Ceph 节点上至少需有两个网卡。

需要对每个 Ceph 节点应用以下更改。对小型集群执行此操作的速度相对较快，但如果集群包含数百甚至数千个节点，则此过程可能十分耗时。

1. 使用以下命令设置集群网络：

   ```
   # ceph config set global cluster_network MY_NETWORK
   ```

   重启动 OSD 以绑定到指定的集群网络：

   ```
   # systemctl restart ceph-*@osd.*.service
   ```

2. 检查专用集群网络是否在 OS 级别按预期工作。

2.1.1.2 不同子网中的监控节点

如果 Monitor 节点位于多个子网中（例如，位于不同的机房并由不同的交换机提供服务），则您需要以 CIDR 表示法指定其公共网络地址。

```
cephuser@adm > ceph config set mon public_network
 "MON_NETWORK_1, MON_NETWORK_2, MON_NETWORK_N
```

例如：

```
cephuser@adm > ceph config set mon public_network "192.168.1.0/24, 10.10.0.0/16"
```

警告

如果按本节所述为公共（或集群）网络指定了多个网络段，则这些子网中的每一个子网都必须能够路由到其他所有子网，否则，不同网络段上的 MON 与其他 Ceph 守护进程将无法通讯，并且集群将随之拆分。此外，如果您使用防火墙，请确保在 iptables 中包含每个 IP 地址或子网，并根据需要在所有节点上为其打开端口。

2.2 多体系结构配置

SUSE Enterprise Storage 支持 x86 和 Arm 体系结构。考虑每个体系结构时，请务必注意从每个 OSD 的内核数、频率和 RAM 的角度而言，不同的 CPU 体系结构在大小调整方面并无实际差异。

与较小的 x86 处理器（非服务器）一样，性能较低的基于 Arm 的内核可能无法提供最佳体验，特别是用于纠删码存储池时。

注意

在整份文档中，使用 `SYSTEM-ARCH` 代替 x86 或 Arm。

2.3 硬件配置

为了获得最佳产品体验，建议您从推荐的集群配置开始。对于测试集群或性能要求较低的集群，我们记录了支持的最低集群配置。

2.3.1 最低集群配置

最低产品集群配置包括：

- 至少四个物理节点（OSD 节点），支持服务共置
- 作为绑定网络的双 10 Gb 以太网
- 一个独立管理节点（可以在外部节点上虚拟化）

详细配置如下：

- 具有 4 GB RAM、四个内核和 1 TB 存储容量的独立管理节点。通常是 Salt 主控端节点。Ceph 服务和网关（例如 Ceph Monitor、元数据服务器、Ceph OSD、对象网关或 NFS Ganesha）在管理节点上不受支持，因为它需要独立地编制集群更新和升级过程。
- 至少四个物理 OSD 节点，每个节点有八个 OSD 磁盘，有关要求请参见第 2.4.1 节 “最低要求”。
  应调整集群总容量的大小，以便即使有一个节点不可用，已用总容量（包括冗余）也不超过 80%。
- 三个 Ceph Monitor 实例。出于延迟原因，需要通过 SSD/NVMe 存储运行 Monitor，而不是通过 HDD 运行。
- Monitor、元数据服务器和网关可以共置在 OSD 节点上，请参见第 2.12 节 “共享一台服务器的 OSD 和 Monitor”以了解 Monitor 共置。如果共置服务，则需要将内存和 CPU 需求相加。
- iSCSI 网关、对象网关和元数据服务器至少需要增量 4 GB RAM 和四个核心。
- 如果您使用的是 CephFS、S3/Swift、iSCSI，则至少需要两个相应角色（元数据服务器、对象网关、iSCSI）的实例才能实现冗余和可用性。
- 这些节点将专用于 SUSE Enterprise Storage，并且不得用于任何其他物理、容器化或虚拟化工作负载。
- 如果 VM 中部署了任何网关（iSCSI、对象网关、NFS Ganesha、元数据服务器等），则这些 VM 不得托管在为其他集群角色提供服务的物理计算机上。（这并非必要，因为它们可作为共置服务受到支持。）
- 在核心物理集群之外的超级管理程序上将服务部署为 VM 时，必须考虑故障域以确保冗余。
  例如，不要在同一个超级管理程序上部署多个同类角色，例如多个 MON 或 MDS 实例。
- 在 VM 内部署时，请务必确保节点具有强大的网络连接性和良好的工作时间同步。
- 超级管理程序节点必须足够大，以避免受到其他占用 CPU、RAM、网络和存储资源的工作负载的干扰。

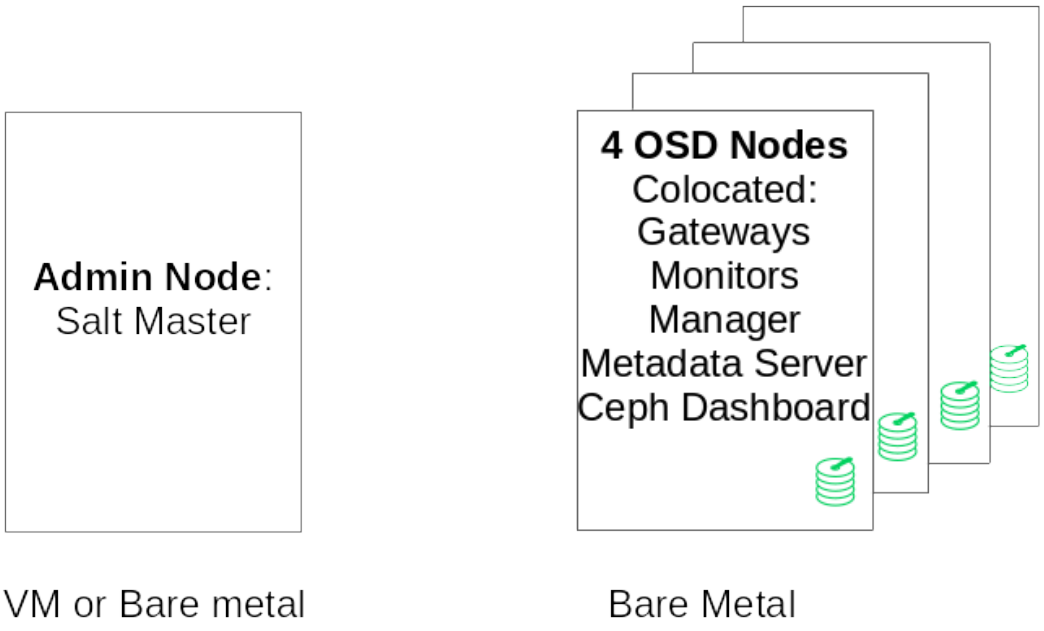

图 2.2： 最低集群配置

2.3.2 建议的生产集群配置

一旦您扩展了集群，建议将 Ceph Monitor、元数据服务器和网关重新定位到不同的节点，以获得更好的容错性。

- 七个对象存储节点
  - 单个节点不超过总存储容量的 15% 左右。
  - 应调整集群总容量的大小，以便即使有一个节点不可用，已用总容量（包括冗余）也不超过 80%。
  - 25 Gb 以太网或以上，进行绑定，一个用于内部集群网络，另一个用于外部公共网络。

- 每个存储集群有 56 个以上的 OSD。
- 有关进一步建议，请参见第 2.4.1 节 “最低要求”。

- 专用的物理基础结构节点。
  - 三个 Ceph Monitor 节点：4 GB RAM，四核处理器，RAID 1 SSD 磁盘。
    有关进一步建议，请参见第 2.5 节 “Monitor 节点”。
  - 对象网关节点：32 GB RAM，八核处理器，RAID 1 SSD 磁盘。
    有关进一步建议，请参见第 2.6 节 “对象网关节点”。
  - iSCSI 网关节点：16 GB RAM，八核处理器，RAID 1 SSD 磁盘。
    有关进一步建议，请参见第 2.9 节 “iSCSI 网关节点”。
  - 元数据服务器节点（一个工作/一个热待机）：32 GB RAM，八核处理器，RAID 1 SSD 磁盘。
    有关进一步建议，请参见第 2.7 节 “元数据服务器节点”。
  - 一个 SES 管理节点：4 GB RAM，四核处理器，RAID 1 SSD 磁盘。

2.3.3 多路径配置

如果要使用多路径硬件，请确保 LVM 在配置文件中的 `devices` 部分下看到 `multipath_component_detection = 1`。可以通过 **`lvm config`** 命令进行此项检查。另外，确保 LVM 通过 LVM 过滤器配置过滤设备的 mpath 组件。此为主机特定。

注意

不建议这样做，只有在无法设置 `multipath_component_detection = 1` 时才应考虑这样做。

有关多路径配置的详细信息，请参见 https://documentation.suse.com/sles/15-SP3/html/SLES-all/cha-multipath.html#sec-multipath-lvm。

2.4 对象存储节点

2.4.1 最低要求

- 以下 CPU 建议针对独立于 Ceph 使用的设备：
  - 每个旋转磁盘 1x 2GHz CPU 线程。
  - 每个 SSD 2x 2GHz CPU 线程。
  - 每个 NVMe 磁盘 4x 2GHz CPU 线程。
- 独立的 10 GbE 网络（公共/客户端和内部），需要 4x 10 GbE，建议 2x 25 GbE。
- 总计所需 RAM = OSD 数量 x (1 GB + `osd_memory_target`) + 16 GB
  有关 `osd_memory_target` 的更多详细信息，请参见《管理和操作指南》, 第 28 章 “Ceph 集群配置”, 第 28.4.1 节 “配置自动缓存大小调整”。
- OSD 磁盘采用 JBOD 配置或单独的 RAID-0 配置。
- OSD 日志可以驻留在 OSD 磁盘上。
- OSD 磁盘应该专门由 SUSE Enterprise Storage 使用。
- 操作系统专用的磁盘和 SSD，最好采用 RAID 1 配置。
- 如果此 OSD 主机将托管用于缓存分层的缓存存储池的一部分，则至少应再额外分配 4 GB RAM。
- Ceph Monitor、网关和元数据服务器可以驻留在对象存储节点上。
- 出于磁盘性能考虑，OSD 节点是裸机节点。OSD 节点上不应运行其他工作负载，除非它是 Ceph Monitor 和 Ceph Manager 的最低设置。
- 根据 6:1 的 SSD 日志与 OSD 的比率为日志提供 SSD。

注意

确保 OSD 节点没有映射任何网络块设备，例如 iSCSI 或 RADOS 块设备映像。

2.4.2 最小磁盘大小

需要在 OSD 上运行两种类型的磁盘空间：用于 WAL/DB 设备的空间以及用于存储数据的主空间。WAL/DB 的最小（默认）值为 6 GB。数据的最小空间为 5 GB，因为系统会自动为小于 5 GB 的分区指定权重 0。

因此，尽管 OSD 的最小磁盘空间为 11 GB，但不建议使用小于 20 GB 的磁盘，即使在测试中也是如此。

2.4.3 BlueStore 的 WAL 和 DB 设备的建议大小

提示：更多信息

有关 BlueStore 的详细信息，请参见第 1.4 节 “BlueStore”。

- 我们建议为 WAL 设备预留 4 GB。虽然 RBD 专用工作站的最小数据库大小为 64 Gb，但建议将对象网关和 CephFS 工作站的数据库大小配置为主设备容量的 2%（但至少为 196 GB）。

  #### 重要

  对于高负载部署，建议使用更大的 DB 卷，特别是在 RGW 或 CephFS 使用率较高的情况下。如果需要，请预留一些容量（槽）来安装更多硬件，以提供更大的 DB 空间。

- 如果您打算将 WAL 和 DB 设备置于同一磁盘，建议您为这两个设备使用一个分区，而不是为每个设备使用单独的分区。这样，Ceph 便可以使用 DB 设备来执行 WAL 操作。这对于磁盘空间的管理也会更有效，因为 Ceph 只会在需要时才会为 WAL 使用 DB 分区。另一个好处是，WAL 分区填满的可能性很小，当该分区未完全利用时，其空间并不会浪费，而是用于 DB 操作。

  要与 WAL 共享 DB 设备，请**不要**指定 WAL 设备，而是仅指定 DB 设备。

  有关指定 OSD 布局的详细信息，请参见《管理和操作指南》, 第 13 章 “操作任务”，第 13.4.3 节 “使用 DriveGroups 规范添加 OSD。”。

2.4.4 用于 WAL/DB 分区的 SSD

固态硬盘 (SSD) 不包含移动部件。这可以减少随机访问时间和读取延迟，同时加快数据吞吐量。由于 SSD 的每 MB 价格大大高于旋转型硬盘，SSD 只适用于较小规模的存储。

如果将 WAL/DB 分区存储在 SSD 上，并将对象数据存储在独立的硬盘上，OSD 的性能会得到大幅提高。

提示：为多个 WAL/DB 分区共享 SSD

由于 WAL/DB 分区占用的空间相对较少，因此可以多个 WAL/DB 分区共享一个 SSD 磁盘。请注意，共享的 WAL/DB 分区每多一个，SSD 磁盘的性能就会相应下降一分。不建议在同一个 SSD 磁盘中共享 6 个以上的 WAL/DB 分区，或者在 NVMe 磁盘中共享 12 个以上的 WAL/DB 分区。

2.4.5 磁盘的最大建议数量

您可以在一台服务器上使用所允许的任意数量的磁盘。规划每台服务器的磁盘数量时，需要考虑以下几点：

- **网络带宽：**在一台服务器中使用的磁盘越多，执行磁盘写入操作时必须通过网卡传输的数据就越多。
- **内存：**系统会将超过 2 GB 的 RAM 用于 BlueStore 缓存。当 `osd_memory_target` 设置为默认值 4 GB 时，该起始缓存大小对于旋转介质而言是比较合理的。如果使用 SSD 或 NVME，请考虑增加缓存大小以及分配给每个 OSD 的 RAM，以便最大限度提高性能。
- **容错：**整台服务器发生故障时，该服务器包含的磁盘越多，则集群暂时丢失的 OSD 就越多。此外，为了确保复制规则的运行，需要将有故障服务器中的所有数据复制到集群中的其他节点。

2.5 Monitor 节点

- 至少需要三个 MON 节点。Monitor 数量应始终为奇数 (1+2n)。
- 4 GB RAM。

- 有四个逻辑内核的处理器。
- 强烈建议对 Monitor 使用 SSD 或其他速度足够快的存储类型，特别是针对每个 Monitor 节点上的 `/var/lib/ceph` 路径，因为仲裁可能不稳定且磁盘延迟较高。建议提供两个采用 RAID 1 配置的磁盘来实现冗余。建议对 Monitor 进程使用独立的磁盘，或者至少是独立的磁盘分区，以防止日志文件缓增等问题导致 Monitor 的可用磁盘空间不足。
- 每个节点只能有一个 Monitor 进程。
- 仅当有足够的硬件资源可用时，才支持混用 OSD、MON 或对象网关节点。这意味着，对于所有服务需要提高相应要求。
- 与多个交换机绑定的两个网络接口。

2.6 对象网关节点

对象网关节点至少应有 6 个 CPU 内核和 32 GB RAM。如果将其他进程共置在同一台计算机上，则需要提高资源的要求。

2.7 元数据服务器节点

元数据服务器节点的适当大小取决于特定用例。一般而言，元数据服务器需要处理的打开文件越多，所需要的 CPU 和 RAM 就越多。以下是最低要求：

- 为每个元数据服务器守护进程分配 4 GB 的 RAM。
- 绑定网络接口。
- 2.5 GHz CPU，至少有两个内核。

2.8 管理节点

至少需要 4 GB RAM 和四核 CPU。其中包括在管理节点上运行 Salt 主控端。对于包含数百个节点的大型集群，建议提供 6 GB RAM。

2.9 iSCSI 网关节点

iSCSI 网关节点至少应有 6 个 CPU 内核和 16 GB RAM。

2.10 SES 和其他 SUSE 产品

本节包含有关将 SES 与其他 SUSE 产品集成的重要信息。

2.10.1 SUSE Manager

SUSE Manager 与 SUSE Enterprise Storage 未集成，因此 SUSE Manager 当前无法管理 SES 集群。

2.11 名称限制

一般情况下，Ceph 不支持在配置文件、存储池名称、用户名等内容中使用非 ASCII 字符。配置 Ceph 集群时，建议在所有 Ceph 对象/配置名称中仅使用简单的字母数字字符（A-Z、a-z、0-9）和最少量的标点符号（“.”、“-”、“_”）。

2.12 共享一台服务器的 OSD 和 Monitor

尽管从技术上讲可在测试环境中的同一台服务器上运行 OSD 和 MON，但强烈建议在生产环境中为每个 Monitor 节点使用独立的服务器。主要原因在于性能 — 集群包含的 OSD 越多，MON 节点需要执行的 I/O 操作就越多。另外，在 MON 节点与 OSD 之间共享一台服务器时，OSD I/O 操作将会成为 Monitor 节点的限制因素。

另一个考虑要点是，是否要在 OSD、MON 节点与服务器上的操作系统之间共享磁盘。答案非常简单：如果可能，请将一个独立的磁盘专门用于 OSD，并将一台独立的服务器用于 Monitor 节点。

尽管 Ceph 支持基于目录的 OSD，但 OSD 应始终包含一个专用磁盘，而不能与操作系统共享一个磁盘。

提示

如果**确实**有必要在同一台服务器上运行 OSD 和 MON 节点，请将一个独立磁盘挂载 `/var/lib/ceph/mon` 目录以在该磁盘上运行 MON，这样可以稍微改善性能。

3 管理节点 HA 设置

管理节点是运行 Salt 主控端服务的 Ceph 集群节点。它管理着集群的其余节点，它会查询这些节点的 Salt 受控端服务并向其发出指示。它通常也会包含其他服务，例如 **Grafana** 仪表盘（由 **Prometheus** 监控工具包提供支持）。

如果管理节点发生故障，通常需要为该节点提供新的工作硬件，并通过最近的备份恢复完整的集群配置堆栈。这种方法很费时，并会导致集群故障。

为防止出现由于管理节点故障导致的 Ceph 集群性能下降，建议您为 Ceph 管理节点使用高可用性 (HA) 集群。

3.1 管理节点的 HA 集群概述

HA 集群的原理是，当其中一个集群节点发生故障时，由另一个节点自动接管其职责，包括虚拟化管理节点。使用此方法时，其他 Ceph 集群节点将不会知道管理节点发生故障。

管理节点的极简 HA 解决方案需要以下硬件：

- 两台能够运行具有高可用性扩展的 SUSE Linux Enterprise 以及虚拟化管理节点的裸机服务器。
- 两个或多个冗余网络通讯路径，例如通过网络设备绑定。
- 用于托管管理节点虚拟机磁盘映像的共享存储。必须能够通过这两台服务器访问共享存储。例如，共享存储可以是 NFS 导出项、Samba 共享或 iSCSI 目标。

有关集群要求的更多详细信息，请访问 https://documentation.suse.com/sle-ha/15-SP3/single-html/SLE-HA-install-quick/#sec-ha-inst-quick-req↗。

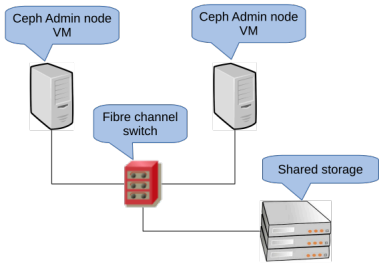

图 3.1： 管理节点的双节点 HA 集群

3.2 构建具有管理节点的 HA 集群

以下过程汇总了构建将管理节点虚拟化的 HA 集群的几个最重要的步骤。有关详细信息，请参见指定链接。

1. 设置一个具有共享存储的基本双节点 HA 集群，如 https://documentation.suse.com/sle-ha/15-SP3/single-html/SLE-HA-install-quick/#art-sleha-install-quick 中所述。
2. 在两个集群节点上，安装运行 KVM 超级管理程序和 `libvirt` 工具包所需的所有软件包，如 https://documentation.suse.com/sles/15-SP3/single-html/SLES-virtualization/#sec-vt-installation-kvm 中所述。
3. 在第一个集群节点上，使用 `libvirt` 创建新的 KVM 虚拟机 (VM)，如 https://documentation.suse.com/sles/15-SP3/single-html/SLES-virtualization/#sec-libvirt-inst-virt-install 中所述。使用预配置的共享存储来存储 VM 的磁盘映像。
4. VM 设置完成后，将其配置导出到共享存储上的 XML 文件。使用以下语法：

```
# virsh dumpxml VM_NAME > /path/to/shared/vm_name.xml
```

5. 为管理节点 VM 创建资源。有关创建 HA 资源的一般信息，请参见 https://documentation.suse.com/sle-ha/15-SP3/single-html/SLE-HA-guide/#cha-conf-hawk2。http://www.linux-ha.org/wiki/VirtualDomain_%28resource_agent%29 中提供了有关为 KVM 虚拟机创建资源的详细信息。

6. 在新创建的 VM guest 中，部署管理节点，包括您需要在其上使用的其他服务。执行第 6 章 “部署 Salt”中的相关步骤。同时，在非 HA 集群服务器上部署其余 Ceph 集群节点。

II 部署 Ceph 集群

4 简介和常见任务 28

5 安装和配置 SUSE Linux Enterprise Server 30

6 部署 Salt 31

7 使用 ceph-salt 部署引导集群 34

8 使用 cephadm 部署其余核心服务 56

9 部署附加服务 73

4 简介和常见任务

自 SUSE Enterprise Storage 7 起，以容器而非 RPM 软件包的形式部署 Ceph 服务。部署过程包含两个基本步骤：

部署引导集群

此阶段称为**第 1 天部署**，包含以下任务：安装底层操作系统，配置 Salt 基础架构，部署由一个 MON 和一个 MGR 服务组成的最小集群。

- 在所有集群节点上安装 SUSE Linux Enterprise Server 15 SP3 并完成底层操作系统的基本配置。
- 在所有集群节点上部署 Salt 基础结构，以便通过 `ceph-salt` 执行初始部署准备。
- 通过 `ceph-salt` 配置集群的基本属性并进行部署。

部署附加服务

在**第 2 天部署**期间，将部署附加核心和非核心 Ceph 服务，例如网关和监控堆栈。

重要

请注意，在初始部署期间，Ceph 社区文档使用 **`cephadm bootstrap`** 命令。`ceph-salt` 会自动调用 **`cephadm bootstrap`** 命令。您不应直接运行 **`cephadm bootstrap`** 命令。不支持使用 **`cephadm bootstrap`** 进行任何手动 Ceph 集群部署。

4.1 阅读发行说明

在发行说明中，可以找到有关自 SUSE Enterprise Storage 的上一个版本发行后所进行的更改的其他信息。检查发行说明以了解：

- 您的硬件是否有特殊注意事项。
- 所使用的任何软件包是否发生了重大更改。
- 是否需要对您的安装实施特殊预防措施。

发行说明还提供未能及时编入手册中的信息。它们还包含有关已知问题的说明。

安装 `release-notes-ses` 软件包之后，可在本地的 `/usr/share/doc/release-notes` 目录中或 https://www.suse.com/releasenotes/ 网页上找到发行说明。

5 安装和配置 SUSE Linux Enterprise Server

1. 在每个集群节点上安装并注册 SUSE Linux Enterprise Server 15 SP3。在安装 SUSE Enterprise Storage 期间，需要访问更新软件源，因此必须注册。至少包含以下模块：

   - Basesystem 模块
   - Server Applications Module

   有关如何安装 SUSE Linux Enterprise Server 的更多详细信息，请参见 https://documentation.suse.com/sles/15-SP3/html/SLES-all/cha-install.html。

2. 在每个集群节点上安装 **SUSE Enterprise Storage 7.1** 扩展。

   ### 提示：将 SUSE Enterprise Storage 与 SUSE Linux Enterprise Server 一起安装

   您可以在安装 SUSE Linux Enterprise Server 15 SP3 之后单独安装 SUSE Enterprise Storage 7.1 扩展，也可以在 SUSE Linux Enterprise Server 15 SP3 安装过程中添加该扩展。

   有关如何安装扩展的更多详细信息，请参见 https://documentation.suse.com/sles/15-SP3/html/SLES-all/cha-register-sle.html。

3. 在每个节点上配置网络设置，包括正确的 DNS 名称解析。有关配置网络的详细信息，请参见 https://documentation.suse.com/sles/15-SP3/single-html/SLES-admin/#sec-network-yast。有关配置 DNS 服务器的详细信息，请参见 https://documentation.suse.com/sles/15-SP3/single-html/SLES-admin/#cha-dns。

6 部署 Salt

SUSE Enterprise Storage 使用 Salt 和 `ceph-salt` 进行初始集群准备。Salt 可帮助您从一个名为 **Salt 主控端**的专用主机同时针对多个集群节点配置和运行命令。在部署 Salt 之前，请考虑以下要点：

- **Salt 受控端**是由一个名为 Salt 主控端的专用节点控制的节点。
- 如果 Salt 主控端主机应属于 Ceph 集群的一部分，则它需要运行自己的 Salt 受控端，但这不是必须的。

提示：每台服务器共享多个角色

如果将每个角色都部署在一个独立节点上，则 Ceph 集群的性能是最佳的。但实际部署有时会要求多个角色共享一个节点。为避免性能欠佳以及升级过程出现问题，请勿向管理节点部署 Ceph OSD、元数据服务器或 Ceph Monitor 角色。

- Salt 受控端需要能通过网络正确解析 Salt 主控端的主机名。默认情况下，受控端 会查找 `salt` 主机名，但您可以在 `/etc/salt/minion` 文件中指定可通过网络访问的任何其他主机名。

1. 在 Salt 主控端节点上安装 `salt-master`：

```
root@master # zypper in salt-master
```

2. 检查 `salt-master` 服务是否已启用并启动，并根据需要将它启用并启动：

```
root@master # systemctl enable salt-master.service
root@master # systemctl start salt-master.service
```

3. 如果要使用防火墙，请确认 Salt 主控端节点是否为所有 Salt 受控端节点打开了端口 4505 和 4506。如果这些端口处于关闭状态，可以使用 **`yast2 firewall`** 命令并通过允许相应区域的 salt-master 服务来打开这些端口。例如，`public`。

4. 在所有受控端节点上安装 `salt-minion` 软件包。

```
root@minion > zypper in salt-minion
```

5. 编辑 `/etc/salt/minion` 并取消注释以下行：

```
#log_level_logfile: warning
```

将 `warning` 日志级别更改为 `info`。

注意：log_level_logfile 和 log_level

`log_level` 用于控制屏幕上将显示的日志消息，而 `log_level_logfile` 用于控制哪些日志消息将写入到 `/var/log/salt/minion`。

注意

请务必更改**所有**集群（受控端）节点上的日志级别。

6. 确保所有其他节点都可以将每个节点的**完全限定域名**解析为公共集群网络上的 IP 地址。

7. 将所有受控端配置为连接到主控端。如果无法通过主机名 `salt` 访问 Salt 主控端，请编辑文件 `/etc/salt/minion`，或创建包含以下内容的新文件 `/etc/salt/minion.d/master.conf`：

```
master: host_name_of_salt_master
```

如果对上述配置文件执行了任何更改，请在所有相关的 Salt 受控端上重启动 Salt 服务：

```
root@minion > systemctl restart salt-minion.service
```

8. 检查所有节点上是否已启用并启动 `salt-minion` 服务。根据需要启用并启动该服务：

```
# systemctl enable salt-minion.service
# systemctl start salt-minion.service
```

9. 确认每个 Salt 受控端的指纹，如果指纹匹配，则接受 Salt 主控端上的所有 Salt 密钥。

注意

如果 Salt 受控端指纹返回空白，请确保 Salt 受控端具有 Salt 主控端配置且可与 Salt 主控端通讯。

查看每个受控端的指纹：

```
root@minion > salt-call --local key.finger
local:
3f:a3:2f:3f:b4:d3:d9:24:49:ca:6b:2c:e1:6c:3f:c3:83:37:f0:aa:87:42:e8:ff...
```

收集到所有 Salt 受控端的指纹后，将列出 Salt 主控端上所有未接受受控端密钥的指纹：

```
root@master # salt-key -F
[...]
Unaccepted Keys:
minion1:
3f:a3:2f:3f:b4:d3:d9:24:49:ca:6b:2c:e1:6c:3f:c3:83:37:f0:aa:87:42:e8:ff...
```

如果受控端的指纹匹配，则接受这些密钥：

```
root@master # salt-key --accept-all
```

10. 校验是否已接受密钥：

```
root@master # salt-key --list-all
```

11. 测试是否所有 Salt 受控端都响应：

```
root@master # salt-run manage.status
```

7 使用 ceph-salt 部署引导集群

本节将指导您完成部署基本 Ceph 集群的过程。请仔细阅读下面的小节，并按给定的顺序执行所包含的命令。

7.1 安装 ceph-salt

`ceph-salt` 提供了用于部署由 cephadm 管理的 Ceph 集群的工具。`ceph-salt` 使用 Salt 基础结构来执行操作系统管理（例如，软件更新或时间同步），并为 Salt 受控端定义角色。

在 Salt 主控端上安装 `ceph-salt` 软件包：

```
root@master # zypper install ceph-salt
```

上面的命令将 `ceph-salt-formula` 安装为依赖项，它通过在 `/etc/salt/master.d` 目录中插入额外的文件修改了 Salt 主控端配置。要应用更改，请重启动 `salt-master.service` 并同步 Salt 模块：

```
root@master # systemctl restart salt-master.service
root@master # salt \* saltutil.sync_all
```

7.2 配置集群属性

使用 **`ceph-salt config`** 命令可配置集群的基本属性。

重要

`/etc/ceph/ceph.conf` 文件由 cephadm 管理，用户**不得**编辑该文件。应使用新的 **`ceph config`** 命令设置 Ceph 配置参数。有关更多信息，请参见《管理和操作指南》，第 28 章 “Ceph 集群配置”，第 28.2 节 “配置数据库”。

7.2.1 使用 ceph-salt 外壳

如果在不带任何路径或子命令的情况下运行 **config**，您将进入交互式 ceph-saltceph-salt 外壳。如果您需要成批配置多个属性，并且不想键入完整的命令语法，使用外壳会非常方便。

```
root@master # ceph-salt config
/> ls
o- / ..................................................................... [...]
  o- ceph_cluster ....................................................... [...]
  | o- minions ..................................................... [no minions]
  | o- roles ....................................................... [...]
  |   o- admin ..................................................... [no minions]
  |   o- bootstrap ................................................. [no minion]
  |   o- cephadm ................................................... [no minions]
  |   o- tuned ..................................................... [...]
  |     o- latency ................................................. [no minions]
  |     o- throughput .............................................. [no minions]
  o- cephadm_bootstrap .................................................. [...]
  | o- advanced ..................................................... [...]
  | o- ceph_conf .................................................... [...]
  | o- ceph_image_path ............................... [ no image path]
  | o- dashboard .................................................... [...]
  | | o- force_password_update ................................ [enabled]
  | | o- password ................................................ [admin]
  | | o- ssl_certificate ....................................... [not set]
  | | o- ssl_certificate_key ................................... [not set]
  | | o- username ................................................ [admin]
  | o- mon_ip ...................................................... [not set]
  o- containers ......................................................... [...]
  | o- registries_conf ..................................... [enabled]
  | | o- registries ................................................ [empty]
  | o- registry_auth ..................................................... [...]
  |   o- password ..................................................... [not set]
  |   o- registry ..................................................... [not set]
  |   o- username ..................................................... [not set]
  o- ssh ..................................................... [no key pair set]
  | o- private_key ................................... [no private key set]
  | o- public_key .................................... [no public key set]
```

```
o- time_server ......................... [enabled, no server host set]
  o- external_servers ....................................... [empty]
  o- servers ................................................ [empty]
  o- subnet ................................................. [not set]
```

从 `ceph-salt` 的 **`ls`** 命令输出中您可以看到，集群配置是以树状结构组织的。要在 `ceph-salt` 外壳中配置集群的特定属性，可使用以下两个选项：

- 从当前位置运行命令，并输入属性的绝对路径作为第一个参数：

```
/> /cephadm_bootstrap/dashboard ls
o- dashboard ..................................................... [...]
  o- force_password_update ..................................... [enabled]
  o- password ....................................................... [admin]
  o- ssl_certificate ........................................... [not set]
  o- ssl_certificate_key ....................................... [not set]
  o- username ....................................................... [admin]
/> /cephadm_bootstrap/dashboard/username set ceph-admin
Value set.
```

- 更改为需要配置和运行命令的属性的路径：

```
/> cd /cephadm_bootstrap/dashboard/
/ceph_cluster/minions> ls
o- dashboard ..................................................... [...]
  o- force_password_update ..................................... [enabled]
  o- password ....................................................... [admin]
  o- ssl_certificate ........................................... [not set]
  o- ssl_certificate_key ....................................... [not set]
  o- username ..................................................[ceph-admin]
```

提示：自动完成配置代码段

在 `ceph-salt` 外壳中，您可以使用类似于常规 Linux 外壳 (Bash) 自动完成的自动完成功能。它可以完成配置路径、子命令或 Salt 受控端名称。自动完成配置路径时，可使用以下两个选项：

- 要让外壳完成您当前位置的相对路径，请按 TAB 键 `→|` 两次。
- 要让外壳完成绝对路径，请输入 `/` 并按 TAB 键 `→|` 两次。

提示：使用光标键导航

如果从 **外壳输入** `cd` `ceph-salt` 而不带任何路径，则该命令将列显集群配置的树状结构，其中当前路径对应的行处于活动状态。您可以使用向上和向下光标键在各行之间导航。按 `Enter` 进行确认后，配置路径将更改为最后一个活动路径。

重要：约定

为了保持文档的一致性，我们将使用单一命令语法，而不输入 `ceph-salt` 外壳。例如，您可以使用以下命令列出集群配置树：

```
root@master # ceph-salt config ls
```

7.2.2 添加 Salt 受控端

将我们在第 6 章 “部署 Salt”中部署并接受的所有或部分 Salt 受控端包含到 Ceph 集群配置中。您可以通过全名指定 Salt 受控端，也可以使用 glob 表达式“*”和“？”一次包含多个 Salt 受控端。在 `/ceph_cluster/minions` 路径下使用 **add** 子命令。以下命令包含所有已接受 Salt 受控端：

```
root@master # ceph-salt config /ceph_cluster/minions add '*'
```

确认是否添加了指定的 Salt 受控端：

```
root@master # ceph-salt config /ceph_cluster/minions ls
o- minions ..................................................... [Minions: 5]
  o- ses-master.example.com ..................................... [no roles]
  o- ses-min1.example.com ....................................... [no roles]
  o- ses-min2.example.com ....................................... [no roles]
  o- ses-min3.example.com ....................................... [no roles]
```

```
  o- ses-min4.example.com ................................. [no roles]
```

7.2.3 指定由 cephadm 管理的 Salt 受控端

指定哪些节点将属于 Ceph 集群并由 cephadm 管理。包含将运行 Ceph 服务的所有节点以及管理节点：

```
root@master # ceph-salt config /ceph_cluster/roles/cephadm add '*'
```

7.2.4 指定管理节点

管理节点是安装 `ceph.conf` 配置文件和 Ceph 管理密钥环的节点。通常在管理节点上运行与 Ceph 相关的命令。

提示：Salt 主控端和管理节点位于同一节点

在所有或大多数主机都属于 SUSE Enterprise Storage 的同类环境中，建议将管理节点和 Salt 主控端置于同一主机。

而对于一个 Salt 基础结构托管多个集群（例如 SUSE Enterprise Storage 和 SUSE Manager）的异构环境，请**勿**将管理节点和 Salt 主控端置于同一主机。

要指定管理节点，请运行以下命令：

```
root@master # ceph-salt config /ceph_cluster/roles/admin add ses-
master.example.com
1 minion added.
root@master # ceph-salt config /ceph_cluster/roles/admin ls
o- admin ................................................ [Minions: 1]
  o- ses-master.example.com ..................... [Other roles: cephadm]
```

提示：在多个节点上安装 ceph.conf 和管理密钥环

如果是部署所需，您可以在多个节点上安装 Ceph 配置文件和管理密钥环。出于安全原因，请避免将其安装在集群的所有节点上。

7.2.5 指定第一个 MON/MGR 节点

您需要指定将要引导集群的集群 Salt 受控端。此受控端将成为第一个运行 Ceph Monitor 和 Ceph Manager 服务的受控端 。

```
root@master # ceph-salt config /ceph_cluster/roles/bootstrap set ses-
min1.example.com
Value set.
root@master # ceph-salt config /ceph_cluster/roles/bootstrap ls
o- bootstrap ..................................... [ses-min1.example.com]
```

此外，您还需要在公共网络上指定引导 MON 的 IP 地址，以确保 `public_network` 参数设置正确，例如：

```
root@master # ceph-salt config /cephadm_bootstrap/mon_ip set 192.168.10.20
```

7.2.6 指定调整的配置

您需要指定集群的哪些受控端具有主动调整的配置。为此，请使用以下命令显式添加这些角色：

注意

一个受控端不能同时拥有 `latency` 和 `throughput` 角色。

```
root@master # ceph-salt config /ceph_cluster/roles/tuned/latency add ses-
min1.example.com
Adding ses-min1.example.com...
1 minion added.
root@master # ceph-salt config /ceph_cluster/roles/tuned/throughput add ses-
min2.example.com
Adding ses-min2.example.com...
1 minion added.
```

7.2.7 生成 SSH 密钥对

cephadm 使用 SSH 协议与集群节点通讯。将自动创建名为 `cephadm` 的用户帐户并用于 SSH 通讯。

您需要生成 SSH 密钥对的私用和公共部分：

```
root@master # ceph-salt config /ssh generate
Key pair generated.
root@master # ceph-salt config /ssh ls
o- ssh ..................................................... [Key Pair set]
  o- private_key ..... [53:b1:eb:65:d2:3a:ff:51:6c:e2:1b:ca:84:8e:0e:83]
  o- public_key ...... [53:b1:eb:65:d2:3a:ff:51:6c:e2:1b:ca:84:8e:0e:83]
```

7.2.8 配置时间服务器

所有集群节点都需要与可靠的时间源进行时间同步。有以下几种方案可以实现时间同步：

- 如果所有集群节点都已配置为使用所选 NTP 服务同步其时间，请完全禁用时间服务器处理：

  ```
  root@master # ceph-salt config /time_server disable
  ```

- 如果您的站点已有单一时间源，请指定时间源的主机名：

  ```
   root@master # ceph-salt config /time_server/servers add time-
  server.example.com
  ```

- 另外，`ceph-salt` 可以配置一个 Salt 受控端来充当集群其余受控端的时间服务器。有时，它被称为“内部时间服务器”。在此方案中，`ceph-salt` 将配置内部时间服务器（应为其中一个 Salt 受控端）以使其时间与外部时间服务器（例如 `pool.ntp.org`）同步，并将所有其他受控端配置为从内部时间服务器获取时间。可以通过以下命令来实现：

  ```
  root@master # ceph-salt config /time_server/servers add ses-
  master.example.com
  root@master # ceph-salt config /time_server/external_servers add
   pool.ntp.org
  ```

`/time_server/subnet` 选项指定允许 NTP 客户端通过其访问 NTP 服务器的子网。当您指定 `/time_server/servers` 时，会自动设置该选项。如果需要更改该选项或手动指定，请运行：

```
root@master # ceph-salt config /time_server/subnet set 10.20.6.0/24
```

检查时间服务器设置：

```
root@master # ceph-salt config /time_server ls
o- time_server ............................................... [enabled]
  o- external_servers ........................................... [1]
  | o- pool.ntp.org ............................................ [...]
  o- servers .................................................... [1]
  | o- ses-master.example.com ................................. [...]
  o- subnet ............................................ [10.20.6.0/24]
```

有关设置时间同步的详细信息，请参见 https://documentation.suse.com/sles/15-SP3/html/SLES-all/cha-ntp.html#sec-ntp-yast。

7.2.9 配置 Ceph Dashboard 登录身份凭证

部署完基本集群后便可使用 Ceph Dashboard。要访问 Ceph Dashboard，您需要设置有效的用户名和密码，例如：

```
root@master # ceph-salt config /cephadm_bootstrap/dashboard/username set admin
root@master # ceph-salt config /cephadm_bootstrap/dashboard/password set PWD
```

提示：强制密码更新

默认情况下，系统将强制第一位仪表盘用户在首次登录仪表盘时更改其密码。要禁用此功能，请运行以下命令：

```
root@master # ceph-salt config /cephadm_bootstrap/dashboard/
force_password_update disable
```

7.2.10 使用容器注册表

Ceph 集群需要能够访问容器注册表，以便能下载和部署容器化 Ceph 服务。可以使用以下两种方式来访问注册表：

- 如果您的集群可以（直接或通过代理）访问 `registry.suse.com` 上的默认注册表，您可以将 `ceph-salt` 直接指向此 URL，而无需创建本地注册表。继续执行第 7.2.10.2 节“配置容器映像的路径”中的步骤。
- 如果集群无法访问默认注册表（例如，若为实体隔离部署），您需要配置本地容器注册表。创建并配置本地注册表后，需要将 `ceph-salt` 指向它。

7.2.10.1 创建并配置本地注册表（可选）

重要

创建本地注册表的方法非常多。本节中的说明是有关创建安全和不安全注册表的示例。有关运行容器映像注册表的一般信息，请参见 https://documentation.suse.com/sles/15-SP3/single-html/SLES-container/#sec-docker-registry-installation。

提示：归置和端口的使用

在集群中的所有节点都可访问的计算机上部署注册表。建议在管理节点上部署。注册表默认会侦听端口 5000。

在注册表节点上，使用以下命令确保端口未被占用：

```
ss -tulpn | grep :5000
```

如果其他进程（例如 `iscsi-tcmu`）已在侦听端口 5000，则确定可用于映射到注册表容器中的端口 5000 的另一个未被占用的端口。

过程 7.1： 创建本地注册表

1. 确认 `Containers Module` 扩展已启用。

```
> SUSEConnect --list-extensions | grep -A2 "Containers Module"
```

```
Containers Module 15 SP3 x86_64 (Activated)
```

2. 确认已安装以下软件包：`apache2-utils`（如果启用了安全注册表）、`cni`、`cni-plugins`、`podman`、`podman-cni-config` 和 `skopeo`。

3. 收集以下信息：

   - 注册表主机的完全限定域名 (`REG_HOST_FQDN`)。
   - 用于映射到注册表容器端口 5000 的可用端口号 (`REG_HOST_PORT`)。
   - 注册表是否安全 (`insecure=[true|false]`)。

4. 要启动不安全的注册表（未经过 SSL 加密），请执行以下步骤：

   a. 为不安全的注册表配置 `ceph-salt`：

   ```
   cephuser@adm > ceph-salt config containers/registries_conf enable
   cephuser@adm > ceph-salt config containers/registries_conf/registries
    \
    add prefix=REG_HOST_FQDN insecure=true \
    location=REG_HOST_PORT:5000
   ```

   b. 启动不安全的注册表，具体做法为创建必要的目录（例如 `/var/lib/registry`）并使用 **`podman`** 命令启动注册表：

   ```
   # mkdir -p /var/lib/registry
   # podman run --privileged -d --name registry \
    -p REG_HOST_PORT:5000 -v /var/lib/registry:/var/lib/registry \
    --restart=always registry:2
   ```

   c. 要使注册表在系统重引导后启动，请为其创建并启用 `systemd` 单元文件：

   ```
   > sudo podman generate systemd --files --name registry
   > sudo mv container-registry.service /etc/systemd/system/
   > sudo systemctl enable container-registry.service
   ```

5. 要启动安全注册表，请执行以下步骤：

   a. 创建必要目录：

```
# mkdir -p /var/lib/registry/{auth,certs}
```

b. 生成 SSL 证书：

```
# openssl req -newkey rsa:4096 -nodes -sha256 \
 -keyout /var/lib/registry/certs/domain.key -x509 -days 365 \
 -out /var/lib/registry/certs/domain.crt
```

注意

将 `Cn=[value]` 的值设置为主机的完全限定域名 ([`REG_HOST_FQDN`])。

c. 将该证书复制到集群的所有节点上并刷新证书缓存：

```
# salt-cp '*' /var/lib/registry/certs/domain.crt \
 /etc/pki/trust/anchors/
# salt '*' cmd.shell "update-ca-certificates"
```

d. 生成用于向注册表进行身份验证的用户名和密码组合：

```
# htpasswd2 -bBc /var/lib/registry/auth/htpasswd \
 REG_USERNAME REG_PASSWORD
```

e. 启动安全注册表。请使用 `REGISTRY_STORAGE_DELETE_ENABLED=true` 标志，以便之后能使用 **`skopeo delete`** 命令删除映像。

```
podman run --name myregistry -p REG_HOST_PORT:5000 \
 -v /var/lib/registry:/var/lib/registry \
 -v /var/lib/registry/auth:/auth:z \
 -e "REGISTRY_AUTH=htpasswd" \
 -e "REGISTRY_AUTH_HTPASSWD_REALM=Registry Realm" \
 -e REGISTRY_AUTH_HTPASSWD_PATH=/auth/htpasswd \
 -v /var/lib/registry/certs:/certs:z \
 -e "REGISTRY_HTTP_TLS_CERTIFICATE=/certs/domain.crt" \
 -e "REGISTRY_HTTP_TLS_KEY=/certs/domain.key" \
 -e REGISTRY_STORAGE_DELETE_ENABLED=true \
 -e REGISTRY_COMPATIBILITY_SCHEMA1_ENABLED=true -d registry:2
```

f. 测试对注册表的安全访问：

```
> curl https://REG_HOST_FQDN:REG_HOST_PORT/v2/_catalog \
 -u REG_USERNAME:REG_PASSWORD
```

6. 创建本地注册表后，需要将 `registry.suse.com` 上官方 SUSE 注册表中的容器映像同步到本地注册表。可以使用 `skopeo` 软件包中的 **`skopeo sync`** 命令来实现此目的。有关详细信息，请参见手册页 (**`man 1 skopeo-sync`**)。考虑下列示例：

例 7.1： 查看清单文件

```
skopeo inspect docker://registry.suse.com/ses/7.1/ceph/ceph | jq .RepoTags
skopeo inspect docker://registry.suse.com/ses/7.1/ceph/grafana |
 jq .RepoTags
skopeo inspect docker://registry.suse.com/ses/7.1/ceph/prometheus-
server:2.32.1 | jq .RepoTags
skopeo inspect docker://registry.suse.com/ses/7.1/ceph/prometheus-node-
exporter:1.1.2 | jq .RepoTags
skopeo inspect docker://registry.suse.com/ses/7.1/ceph/prometheus-
alertmanager:0.21.0 | jq .RepoTags
```

例 7.2： 同步到目录

同步所有 Ceph 映像：

```
skopeo sync --src docker --dest dir registry.suse.com/ses/7.1/ceph/
ceph /root/images/
```

仅同步最新的映像：

```
skopeo sync --src docker --dest dir registry.suse.com/ses/7.1/ceph/
ceph:latest /root/images/
```

例 7.3： 同步 GRAFANA 映像：

```
skopeo sync --src docker --dest dir registry.suse.com/ses/7.1/ceph/
grafana /root/images/
```

仅同步最新的 Grafana 映像：

```
skopeo sync --src docker --dest dir registry.suse.com/ses/7.1/ceph/
grafana:latest /root/images/
```

例 7.4： 同步最新的 PROMETHEUS 映像

```
skopeo sync --src docker --dest dir registry.suse.com/ses/7.1/ceph/
prometheus-server:2.32.1 /root/images/
skopeo sync --src docker --dest dir registry.suse.com/ses/7.1/ceph/
prometheus-node-exporter:1.1.2 /root/images/
skopeo sync --src docker --dest dir registry.suse.com/ses/7.1/ceph/
prometheus-alertmanager:0.21.0 /root/images/
```

过程 7.2： 配置本地注册表和访问身份凭证

1. 配置本地注册表的 URL：

```
cephuser@adm > ceph-salt config /containers/registry_auth/registry
 set REG_HOST_URL
```

2. 配置用户名和密码以访问本地注册表：

```
cephuser@adm > ceph-salt config /containers/registry_auth/username
 set REG_USERNAME
```

```
cephuser@adm > ceph-salt config /containers/registry_auth/password
 set REG_PASSWORD
```

提示：注册表缓存

要避免在出现新的更新容器时重新同步本地注册表，您可以配置**注册表缓存**。

7.2.10.2 配置容器映像的路径

重要

本节帮助您配置引导集群（部署的第一个 Ceph Monitor 和 Ceph Manager 对）的容器映像的路径。该路径不适用于附加服务（如监控堆栈）的容器映像。

提示：配置 HTTPS 代理

如果您需要使用代理来与容器注册表服务器通讯，请在所有集群节点上执行以下配置步骤：

1. 复制容器的配置文件：

```
> sudo cp /usr/share/containers/containers.conf /etc/containers/
containers.conf
```

2. 编辑新复制的文件，在文件的 `[engine]` 部分添加 `http_proxy` 设置，例如：

```
> cat /etc/containers/containers.conf
 [engine]
 http_proxy=proxy.example.com
 [...]
```

cephadm 需要知道容器映像的有效 URI 路径。执行以下命令校验默认设置

```
root@master # ceph-salt config /cephadm_bootstrap/ceph_image_path ls
```

如果您不需要备用或本地注册表，请指定默认的 SUSE 容器注册表：

```
root@master # ceph-salt config /cephadm_bootstrap/ceph_image_path set
 registry.suse.com/ses/7.1/ceph/ceph
```

如果您的部署需要特定路径（例如本地注册表的路径），请按如下所示进行配置：

```
root@master # ceph-salt config /cephadm_bootstrap/ceph_image_path
 set LOCAL_REGISTRY_PATH
```

7.2.11 启用数据传输中加密 (msgr2)

Messenger v2 协议 (MSGR2) 是 Ceph 的线上传输协议。它提供了一种可对正在通过网络传输的所有数据进行加密的安全模式，封装了身份验证有效负载，并支持未来集成新的身份验证模式（例如 Kerberos）。

重要

Linux 内核 Ceph 客户端（例如 CephFS 和 RADOS 块设备）当前不支持 msgr2。

Ceph 守护进程可以绑定到多个端口，以便让旧 Ceph 客户端和支持 v2 的新客户端能够连接到同一集群。默认情况下，针对新的 v2 协议，MON 现在会绑定到 IANA 指定的新端口 3300（CE4h 或 0xCE4），而针对旧版 v1 协议，则会绑定到旧的默认端口 6789。

v2 协议 (MSGR2) 支持以下两种连接模式：

crc 模式

建立连接时进行强初始身份验证和 CRC32C 完整性检查。

secure 模式

建立连接时进行强初始身份验证，并对所有身份验证后流量进行完全加密，包括加密完整性检查。

对于大多数连接，有一些选项可以控制使用哪种模式：

ms_cluster_mode

用于 Ceph 守护进程之间的集群内通讯的连接模式（或允许的模式）。如果列出了多种模式，则首选最先列出的模式。

ms_service_mode

连接到集群时允许客户端使用的模式列表。

ms_client_mode

与 Ceph 集群通讯时，供客户端使用（或允许）的连接模式列表，按首选项排序。

有一组专用于 Monitor 的并行选项集，允许管理员设置与 Monitor 通讯的不同（通常更安全）要求。

ms_mon_cluster_mode

在 Monitor 之间使用的连接模式（或允许的模式）。

ms_mon_service_mode

连接到 Monitor 时，供客户端或其他 Ceph 守护进程使用的允许模式列表。

ms_mon_client_mode

连接到 Monitor 时，供客户端或非 Monitor 守护进程使用的连接模式的列表，按首选项排序。

要在部署期间启用 MSGR2 加密模式，您需要在运行 `ceph-salt apply` **之前向** ceph-salt 配置添加一些配置选项。

要使用 `secure` 模式，请运行以下命令。

向 配置工具中的 ceph_conf`ceph-salt` 添加全局部分。

```
root@master # ceph-salt config /cephadm_bootstrap/ceph_conf add global
```

设置以下选项：

```
root@master # ceph-salt config /cephadm_bootstrap/ceph_conf/global set
 ms_cluster_mode "secure crc"
root@master # ceph-salt config /cephadm_bootstrap/ceph_conf/global set
 ms_service_mode "secure crc"
root@master # ceph-salt config /cephadm_bootstrap/ceph_conf/global set
 ms_client_mode "secure crc"
```

注意

确保 `secure` 先于 `crc`。

要**强制使用** `secure` 模式，请运行以下命令：

```
root@master # ceph-salt config /cephadm_bootstrap/ceph_conf/global set
 ms_cluster_mode secure
root@master # ceph-salt config /cephadm_bootstrap/ceph_conf/global set
 ms_service_mode secure
root@master # ceph-salt config /cephadm_bootstrap/ceph_conf/global set
 ms_client_mode secure
```

提示：更新设置

如果您要更改上述任何设置，请在 Monitor 配置存储中设置配置更改。此操作使用 **`ceph config set`** 命令实现。

```
root@master # ceph config set global CONNECTION_OPTION CONNECTION_MODE [--
force]
```

例如：

```
root@master # ceph config set global ms_cluster_mode "secure crc"
```

如果要检查当前值（包括默认值），请运行以下命令：

```
root@master # ceph config get CEPH_COMPONENT CONNECTION_OPTION
```

例如，要获取 OSD 的 `ms_cluster_mode`，请运行：

```
root@master # ceph config get osd ms_cluster_mode
```

7.2.12 配置集群网络

（可选）如果运行的是独立的集群网络，则可能需要设置集群网络 IP 地址并后跟斜线符号及子网掩码部分，例如 `192.168.10.22/24`。

运行以下命令可启用 `cluster_network`：

```
root@master # ceph-salt config /cephadm_bootstrap/ceph_conf add global
root@master # ceph-salt config /cephadm_bootstrap/ceph_conf/global set
 cluster_network NETWORK_ADDR
```

7.2.13 确认集群配置

最低集群配置已完成。请检查是否存在明显错误：

```
root@master # ceph-salt config ls
```

```
o- / ..................................................... [...]
  o- ceph_cluster ....................................... [...]
  | o- minions ..................................... [Minions: 5]
  | | o- ses-master.example.com ..................... [admin]
  | | o- ses-min1.example.com ..... [bootstrap, admin]
  | | o- ses-min2.example.com ..... [no roles]
  | | o- ses-min3.example.com ..... [no roles]
  | | o- ses-min4.example.com ..... [no roles]
  | o- roles ..................................... [...]
  |   o- admin ..................................... [Minions: 2]
  |   | o- ses-master.example.com ..... [no other roles]
  |   | o- ses-min1.example.com ..... [other roles: bootstrap]
  |   o- bootstrap ..... [ses-min1.example.com]
  |   o- cephadm ..... [Minions: 5]
  |   o- tuned ..... [...]
  |     o- latency ..... [no minions]
  |     o- throughput ..... [no minions]
  o- cephadm_bootstrap ..... [...]
  | o- advanced ..... [...]
  | o- ceph_conf ..... [...]
  | o- ceph_image_path ..... [registry.suse.com/ses/7.1/ceph/ceph]
  | o- dashboard ..... [...]
  |   o- force_password_update ..... [enabled]
  |   o- password ..... [randomly generated]
  |   o- username ..... [admin]
  | o- mon_ip ..... [192.168.10.20]
  o- containers ..... [...]
  | o- registries_conf ..... [enabled]
  | | o- registries ..... [empty]
  | o- registry_auth ..... [...]
  |   o- password ..... [not set]
  |   o- registry ..... [not set]
  |   o- username ..... [not set]
  o- ssh ..... [Key Pair set]
  | o- private_key ..... [53:b1:eb:65:d2:3a:ff:51:6c:e2:1b:ca:84:8e:0e:83]
  | o- public_key ..... [53:b1:eb:65:d2:3a:ff:51:6c:e2:1b:ca:84:8e:0e:83]
  o- time_server ..... [enabled]
    o- external_servers ..... [1]
```

```
    | o- 0.pt.pool.ntp.org ..................................... [...]
    o- servers ....................................................... [1]
    | o- ses-master.example.com ..................................... [...]
    o- subnet ..................................... [10.20.6.0/24]
```

提示：集群配置的状态

您可以通过运行以下命令检查集群的配置是否有效：

```
root@master # ceph-salt status
cluster: 5 minions, 0 hosts managed by cephadm
config: OK
```

7.2.14 导出集群配置

在配置了基本集群并且其配置有效之后，最好将其配置导出到文件中：

```
root@master # ceph-salt export > cluster.json
```

警告

ceph-salt export 的输出中包含 SSH 私用密钥。如果您担心安全问题，请不要在未采取适当预防措施的情况下执行此命令。

如果集群配置损坏并需要恢复到备份状态，请运行：

```
root@master # ceph-salt import cluster.json
```

7.3 更新节点并引导最小集群

部署集群之前，请更新所有节点上的所有软件包：

```
root@master # ceph-salt update
```

如果节点在更新期间报告 `Reboot is needed`，则表示重要的操作系统软件包（例如内核）已更新到更新版本，您需要重引导节点才能应用更改。

要重引导所有需要重引导的节点，请追加 `--reboot` 选项

```
root@master # ceph-salt update --reboot
```

或者在单独的步骤中重引导这些节点：

```
root@master # ceph-salt reboot
```

重要

永远不会通过 **`ceph-salt update --reboot`** 或 **`ceph-salt reboot`** 命令重引导 Salt 主控端。如果 Salt 主控端需要重引导，您需要手动进行重引导。

更新节点后，请引导最小集群：

```
root@master # ceph-salt apply
```

注意

引导完成后，集群将拥有一个 Ceph Monitor 和一个 Ceph Manager。

以上命令将打开一个交互式用户界面，其中会显示每个受控端的当前进度。

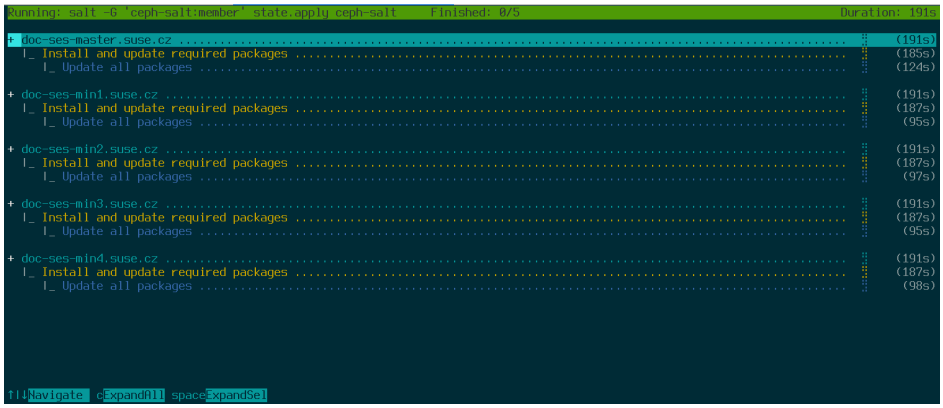

图 7.1： 部署最小集群

提示：非交互模式

如果您需要从脚本应用配置，还有一种非交互的部署模式。此模式在从远程计算机部署集群时也很有用，因为通过网络不断更新屏幕上的进度信息可能会对用户造成干扰：

```
root@master # ceph-salt apply --non-interactive
```

7.4 查看最后的步骤

在 **`ceph-salt apply`** 命令完成后，您便应该会有一个 Ceph Monitor 和一个 Ceph Manager。您应该能够在任何以 `root` 身份或使用 `sudo` 的 `cephadm` 用户身份被授予 `admin` 角色的受控端上成功运行 **`ceph status`** 命令。

后续步骤包括使用 cephadm 部署额外的 Ceph Monitor、Ceph Manager、OSD、监控堆栈和网关。

继续之前，请查看新集群的网络设置。此时，已根据在 `ceph-salt` 配置中针对 `/cephadm_bootstrap/mon_ip` 输入的值填充了 `public_network` 设置。不过，此设置仅适用于 Ceph Monitor。您可以使用以下命令查看此设置：

```
root@master # ceph config get mon public_network
```

这是 Ceph 正常工作所需的最低配置，但还是建议您将此 `public_network` 设为 `global`，这意味着它将应用于所有类型的 Ceph 守护进程，而不仅仅应用于 MON：

```
root@master # ceph config set global public_network "$(ceph config get mon
 public_network)"
```

注意

此步骤不是必需的。但如果不使用此设置，Ceph OSD 和其他守护进程（Ceph Monitor 除外）将侦听**所有地址**。

如果您希望在各 OSD 之间使用完全独立的网络进行通讯，请运行以下命令：

```
root@master # ceph config set global cluster_network
 "cluster_network_in_cidr_notation"
```

执行此命令将确保您部署中所创建的 OSD 从一开始就使用预期的集群网络。

如果您的集群设置为具有密集节点（每个主机有超过 62 个 OSD），请确保为 Ceph OSD 指定足够的端口。默认范围 (6800-7300) 当前允许每个主机有不超过 62 个 OSD。对于具有密集节点的集群，请将设置 `ms_bind_port_max` 调整到适当的值。每个 OSD 将使用 8 个额外的端口。例如，如果一台主机设置为运行 96 个 OSD，则需要 768 个端口。通过运行以下命令，应将 `ms_bind_port_max` 至少设置为 7568：

```
root@master # ceph config set osd.* ms_bind_port_max 7568
```

您需要相应地调整防火墙设置才能使其正常工作。有关更多信息，请参见《Troubleshooting Guide》, 第 13 章 “Hints and tips”, 第 13.7 节 “Firewall settings for Ceph”。

7.5 禁用不安全的客户端

从 Pacific v15.2.11 起，引入了新的健康状况警告来告知您允许了不安全的客户端加入集群。此警告默认会**打开**。Ceph Dashboard 会表明集群处于 `HEALTH_WARN` 状态，在命令行上校验集群状态会告知您如下信息：

```
cephuser@adm > ceph status
cluster:
  id:     3fe8b35a-689f-4970-819d-0e6b11f6707c
  health: HEALTH_WARN
  mons are allowing insecure global_id reclaim
[...]
```

此警告表示 Ceph Monitor 仍在允许未增补的旧版客户端连接集群。这样可确保在升级集群时，现有客户端仍可连接集群，但会警告您存在需要解决的问题。当集群和所有客户端都升级到最新版本的 Ceph 后，运行以下命令禁用未增补的客户端：

```
cephuser@adm > ceph config set mon auth_allow_insecure_global_id_reclaim false
```

8 使用 cephadm 部署其余核心服务

部署基本 Ceph 集群之后，请将核心服务部署到更多集群节点。为使客户端能够访问集群数据，还需要部署额外的服务。

目前，我们支持使用 Ceph orchestrator（**ceph orch** 子命令）在命令行上部署 Ceph 服务。

8.1 ceph orch 命令

Ceph orchestrator 命令 **ceph orch** 是 cephadm 模块的接口，它负责列出集群组件并在新的集群节点上部署 Ceph 服务。

8.1.1 显示 orchestrator 状态

以下命令会显示当前模式和 Ceph orchestrator 的状态。

```
cephuser@adm > ceph orch status
```

8.1.2 列出设备、服务和守护进程

要列出所有磁盘设备，请运行以下命令：

```
cephuser@adm > ceph orch device ls
Hostname   Path      Type  Serial  Size   Health   Ident  Fault  Available
ses-master /dev/vdb  hdd   0d8a... 10.7G  Unknown  N/A    N/A    No
ses-min1   /dev/vdc  hdd   8304... 10.7G  Unknown  N/A    N/A    No
ses-min1   /dev/vdd  hdd   7b81... 10.7G  Unknown  N/A    N/A    No
[...]
```

提示：服务和守护进程

服务是表示特定类型的 Ceph 服务的一种通用术语，例如 Ceph Manager。

守护进程表示服务的特定实例，例如在名为 `ses-min1` 的节点上运行的进程 `mgr.ses-min1.gdlcik`。

要列出 cephadm 已知的所有服务，请运行：

```
cephuser@adm > ceph orch ls
NAME  RUNNING  REFRESHED  AGE  PLACEMENT  IMAGE NAME                  IMAGE ID
mgr       1/0  5m ago     -    <no spec>  registry.example.com/[...]
 5bf12403d0bd
mon       1/0  5m ago     -    <no spec>  registry.example.com/[...]
 5bf12403d0bd
```

提示

您可以使用可选的 `--host` 参数使列表仅显示特定节点上的服务，也可使用可选的 `--service-type` 参数使列表仅显示特定类型的服务。可接受的类型有 `mon`、`osd`、`mgr`、`mds` 和 `rgw`。

要列出由 cephadm 部署的所有运行中守护进程，请运行：

```
cephuser@adm > ceph orch ps
NAME            HOST     STATUS   REFRESHED AGE VERSION    IMAGE ID
 CONTAINER ID
mgr.ses-min1.gd ses-min1 running) 8m ago    12d 15.2.0.108 5bf12403d0bd
 b8104e09814c
mon.ses-min1    ses-min1 running) 8m ago    12d 15.2.0.108 5bf12403d0bd
 a719e0087369
```

提示

要查询某个特定守护进程的状态，请使用 `--daemon_type` 和 `--daemon_id`。对于 OSD，ID 为数字 OSD ID。对于 MDS，ID 为文件系统名称：

```
cephuser@adm > ceph orch ps --daemon_type osd --daemon_id 0
cephuser@adm > ceph orch ps --daemon_type mds --daemon_id my_cephfs
```

8.2 服务和归置规范

指定 Ceph 服务部署的推荐方法是创建一个 YAML 格式的文件，其中包含所要部署服务的规范。

8.2.1 创建服务规范

您可以为每种类型的服务创建单独的规范文件，例如：

```
root@master # cat nfs.yml
service_type: nfs
service_id: EXAMPLE_NFS
placement:
  hosts:
  - ses-min1
  - ses-min2
spec:
  pool: EXAMPLE_POOL
  namespace: EXAMPLE_NAMESPACE
```

或者，您可以在一个描述哪些节点将运行特定服务的文件（例如 `cluster.yml`）中指定多个（或所有）服务类型。请记得用三个破折号 (`---`) 分隔各个服务类型：

```
cephuser@adm > cat cluster.yml
service_type: nfs
service_id: EXAMPLE_NFS
placement:
  hosts:
  - ses-min1
  - ses-min2
spec:
  pool: EXAMPLE_POOL
  namespace: EXAMPLE_NAMESPACE
---
service_type: rgw
service_id: REALM_NAME.ZONE_NAME
placement:
  hosts:
```

```
  - ses-min1
  - ses-min2
  - ses-min3
---
[...]
```

上述属性的含义如下：

`service_type`
: 服务的类型。它可以是 Ceph 服务（`mon`、`mgr`、`mds`、`crash`、`osd` 或 `rbd-mirror`）、网关（`nfs` 或 `rgw`）或部分监控堆栈（`alertmanager`、`grafana`、`node-exporter` 或 `prometheus`）。

`service_id`
: 服务的名称。`mon`、`mgr`、`alertmanager`、`grafana`、`node-exporter` 和 `prometheus` 类型的规范不需要 `service_id` 属性。

`placement`
: 指定哪些节点将运行服务。有关更多详细信息，请参见第 8.2.2 节 “创建归置规范”。

`spec`
: 与服务类型相关的其他规范。

提示：应用特定服务

Ceph 集群服务通常具有许多专属属性。有关各个服务规范的示例和详细信息，请参见第 8.3 节 “部署 Ceph 服务”。

8.2.2 创建归置规范

要部署 Ceph 服务，cephadm 需要知道要在其上部署这些服务的节点。请使用 `placement` 属性并列出要应用服务的节点的主机名简短名称：

```
cephuser@adm > cat cluster.yml
[...]
placement:
```

```
  hosts:
  - host1
  - host2
  - host3
[...]
```

8.2.3 应用集群规范

创建包含所有服务及其归置的规范的完整 `cluster.yml` 文件之后，您便可通过运行以下命令来应用集群：

```
cephuser@adm > ceph orch apply -i cluster.yml
```

要查看集群的状态，请运行 **`ceph orch status`** 命令。有关细节，请参见第 8.1.1 节 “显示 orchestrator 状态”。

8.2.4 导出运行中集群的规范

虽然您使用如第 8.2 节 “服务和归置规范”中所述的规范文件将服务部署到 Ceph 集群，但在实际操作过程中集群的配置可能会偏离原始规范。另外，您也可能会无意间删除规范文件。

要检索运行中集群的完整规范，请运行：

```
cephuser@adm > ceph orch ls --export
placement:
  hosts:
  - hostname: ses-min1
    name: ''
    network: ''
service_id: my_cephfs
service_name: mds.my_cephfs
service_type: mds
---
placement:
  count: 2
service_name: mgr
```

```
service_type: mgr
---
[...]
```

提示

您可以追加 `--format` 选项以更改默认的 `yaml` 输出格式。您可以从 `json`、`json-pretty` 或 `yaml` 中进行选择。例如：

```
ceph orch ls --export --format json
```

8.3 部署 Ceph 服务

在运行基本集群之后，您可以将 Ceph 服务部署到其他节点。

8.3.1 部署 Ceph Monitor 和 Ceph Manager

Ceph 集群的不同节点之间部署了三个或五个 MON。如果集群中有五个或更多节点，则建议部署五个 MON。好的做法是，将 MGR 部署在与 MON 相同的节点上。

重要：包含引导 MON

在部署 MON 和 MGR 时，请记得包含在第 7.2.5 节 “指定第一个 MON/MGR 节点”中配置基本集群时添加的第一个 MON。

要部署 MON，请应用以下规范：

```
service_type: mon
placement:
  hosts:
  - ses-min1
  - ses-min2
  - ses-min3
```

注意

如果需要添加另一个节点，请在同一 YAML 列表中追加主机名。例如：

```
service_type: mon
placement:
 hosts:
 - ses-min1
 - ses-min2
 - ses-min3
 - ses-min4
```

同样，要部署 MGR，请应用以下规范：

重要

确保您的部署在每个部署中至少有三个 Ceph Manager。

```
service_type: mgr
placement:
  hosts:
  - ses-min1
  - ses-min2
  - ses-min3
```

提示

如果 MON 或 MGR **不**在同一子网中，则需要追加子网地址。例如：

```
service_type: mon
placement:
  hosts:
  - ses-min1:10.1.2.0/24
  - ses-min2:10.1.5.0/24
  - ses-min3:10.1.10.0/24
```

8.3.2 部署 Ceph OSD

重要：当存储设备可用时

如果满足以下所有条件，则存储设备将被视为**可用**：

- 设备没有分区。
- 设备没有任何 LVM 状态。
- 设备未挂载。
- 设备不包含文件系统。
- 设备不包含 BlueStore OSD。
- 设备大于 5 GB。

如果不符合上述条件，Ceph 将拒绝供给此类 OSD。

可以使用以下两种方式来部署 OSD：

- 告知 Ceph 使用所有可用和未使用的存储设备：

```
cephuser@adm > ceph orch apply osd --all-available-devices
```

- 使用 DriveGroups（参见《管理和操作指南》, 第 13 章 “操作任务”, 第 13.4.3 节 “使用 DriveGroups 规范添加 OSD。”）创建 OSD 规范，该规范描述了将根据设备属性（例如设备类型（SSD 或 HDD）、设备型号名称、大小或设备所在的节点）部署的设备。然后通过运行以下命令应用规范：

```
cephuser@adm > ceph orch apply osd -i drive_groups.yml
```

8.3.3 部署元数据服务器

CephFS 需要一个或多个元数据服务器 (MDS) 服务。要创建 CephFS，首先要通过应用以下规范来创建 MDS 服务器：

注意

在应用以下规范之前，请确保至少创建了两个存储池，一个用于存储 CephFS 数据，另一个用于存储 CephFS 元数据。

```
service_type: mds
service_id: CEPHFS_NAME
placement:
  hosts:
  - ses-min1
  - ses-min2
  - ses-min3
```

MDS 正常运行后，创建 CephFS：

```
ceph fs new CEPHFS_NAME metadata_pool data_pool
```

8.3.4 部署对象网关

cephadm 会将对象网关部署为管理特定**领域**和**区域**的守护进程的集合。

您可以将对象网关服务与现有的领域和区域相关联（有关更多详细信息，请参见《管理和操作指南》, 第 21 章 “Ceph 对象网关”, 第 21.13 节 “多站点对象网关”），也可以指定不存在的 REALM_NAME 和 ZONE_NAME，应用以下配置后会自动创建相应领域和区域：

```
service_type: rgw
service_id: REALM_NAME.ZONE_NAME
placement:
  hosts:
  - ses-min1
  - ses-min2
  - ses-min3
spec:
  rgw_realm: RGW_REALM
  rgw_zone: RGW_ZONE
```

8.3.4.1 使用安全 SSL 访问

要使用到对象网关的安全 SSL 连接，您需要一对有效的 SSL 证书和密钥文件（有关更多详细信息，请参见《管理和操作指南》, 第 21 章 “Ceph 对象网关”, 第 21.7 节 “为对象网关启用 HTTPS/SSL”）。您需要启用 SSL，指定 SSL 连接的端口号以及 SSL 证书和密钥文件。

要启用 SSL 并指定端口号，请在规范中包含以下内容：

```
spec:
  ssl: true
  rgw_frontend_port: 443
```

要指定 SSL 证书和密钥，可以将其内容直接粘贴到 YAML 规范文件中。行末的竖线符号 (|) 告知分析程序预期的值为多行字符串。例如：

```
spec:
  ssl: true
  rgw_frontend_port: 443
  rgw_frontend_ssl_certificate: |
   -----BEGIN CERTIFICATE-----
   MIIFmjCCA4KgAwIBAgIJAIZ2n35bmwXTMA0GCSqGSIb3DQEBCwUAMGIxCzAJBgNV
   BAYTAkFVMQwwCgYDVQQIDANOU1cxHTAbBgNVBAoMFEV4YW1wbGUgUkdXIFNTTCBp
   [...]
   -----END CERTIFICATE-----
   rgw_frontend_ssl_key: |
   -----BEGIN PRIVATE KEY-----
   MIIJRAIBADANBgkqhkiG9w0BAQEFAASCCS4wggkqAgEAAoICAQDLtFwg6LLl2j4Z
   BDV+iL4AO7VZ9KbmWIt37Ml2W6y2YeKX3Qwf+3eBz7TVHR1dm6iPpCpqpQjXUsT9
   [...]
   -----END PRIVATE KEY-----
```

提示

您可以省略 `rgw_frontend_ssl_certificate:` 和 `rgw_frontend_ssl_key:` 关键字，并将它们上载到配置数据库，而不是粘贴 SSL 证书和密钥文件的内容：

```
cephuser@adm > ceph config-key set rgw/cert/REALM_NAME/ZONE_NAME.crt \
 -i SSL_CERT_FILE
cephuser@adm > ceph config-key set rgw/cert/REALM_NAME/ZONE_NAME.key \
```

```
 -i SSL_KEY_FILE
```

8.3.4.1.1 将对象网关配置为同时侦听端口 443 和 80

要将对象网关配置为同时侦听端口 443 (HTTPS) 和 80 (HTTP)，请执行以下步骤：

注意

该过程中的命令使用 `default` 领域和区域。

1. 提供规范文件以部署对象网关。有关对象网关规范的更多详细信息，请参见第 8.3.4 节 “部署对象网关”。使用以下命令：

```
cephuser@adm > ceph orch apply -i SPEC_FILE
```

2. 如果规范文件中未提供 SSL 证书，请使用以下命令添加证书：

```
cephuser@adm > ceph config-key set rgw/cert/default/default.crt -i
 certificate.pem
cephuser@adm > ceph config-key set rgw/cert/default/default.key -i key.pem
```

3. 更改 `rgw_frontends` 选项的默认值：

```
cephuser@adm > ceph config set client.rgw.default.default rgw_frontends \
 "beast port=80 ssl_port=443"
```

4. 删除 cephadm 创建的特定配置。运行以下命令标识 `rgw_frontends` 选项是为哪个目标配置的：

```
cephuser@adm > ceph config dump | grep rgw
```

例如，目标是 `client.rgw.default.default.node4.yiewdu`。删除 `rgw_frontends` 当前的具体值：

```
cephuser@adm > ceph config rm client.rgw.default.default.node4.yiewdu
 rgw_frontends
```

提示

您也可以不删除 `rgw_frontends` 的值，而是指定其值。例如：

```
cephuser@adm > ceph config set
 client.rgw.default.default.node4.yiewdu \
 rgw_frontends "beast port=80 ssl_port=443"
```

5. 重启动对象网关：

```
cephuser@adm > ceph orch restart rgw.default.default
```

8.3.4.2 使用子集群部署

子集群可帮助您组织集群中的节点，以分隔工作负载，让弹性缩放更轻松。如果使用子集群进行部署，请应用以下配置：

```
service_type: rgw
service_id: REALM_NAME.ZONE_NAME.SUBCLUSTER
placement:
  hosts:
  - ses-min1
  - ses-min2
  - ses-min3
spec:
  rgw_realm: RGW_REALM
  rgw_zone: RGW_ZONE
  subcluster: SUBCLUSTER
```

8.3.5 部署 iSCSI 网关

cephadm 可部署 iSCSI 网关，它是一种存储区域网络 (SAN) 协议，可让客户端（称作发起程序）将 SCSI 命令发送到远程服务器上的 SCSI 存储设备（目标）。

应用以下配置进行部署。确保 `trusted_ip_list` 包含所有 iSCSI 网关和 Ceph Manager 节点的 IP 地址（请参见下面的输出示例）。

注意

请确保在应用以下规范之前创建存储池。

```
service_type: iscsi
service_id: EXAMPLE_ISCSI
placement:
  hosts:
  - ses-min1
  - ses-min2
  - ses-min3
spec:
  pool: EXAMPLE_POOL
  api_user: EXAMPLE_USER
  api_password: EXAMPLE_PASSWORD
  trusted_ip_list: "IP_ADDRESS_1,IP_ADDRESS_2"
```

注意

请确保针对 `trusted_ip_list` 列出的 IP 在逗号分隔后**没有**空格。

8.3.5.1 安全 SSL 配置

要在 Ceph Dashboard 和 iSCSI 目标 API 之间使用安全 SSL 连接，您需要一对有效的 SSL 证书和密钥文件。它们可以是 CA 颁发的，也可以是自我签名的（参见《管理和操作指南》, 第 10 章 “手动配置”, 第 10.1.1 节 “创建自我签名证书”）。要启用 SSL，请在您的规范文件中包含 `api_secure: true` 设置：

```
spec:
  api_secure: true
```

要指定 SSL 证书和密钥，可以将其内容直接粘贴到 YAML 规范文件中。行末的竖线符号 (`|`) 告知分析程序预期的值为多行字符串。例如：

```
spec:
  pool: EXAMPLE_POOL
  api_user: EXAMPLE_USER
  api_password: EXAMPLE_PASSWORD
  trusted_ip_list: "IP_ADDRESS_1,IP_ADDRESS_2"
  api_secure: true
  ssl_cert: |
    -----BEGIN CERTIFICATE-----
    MIIDtTCCAp2gAwIBAgIYMC4xNzc1NDQxNjEzMzc2MjMyXzxvQ7EcMA0GCSqGSIb3
    DQEBCwUAMG0xCzAJBgNVBAYTAlVTMQ0wCwYDVQQIDARVdGFoMRcwFQYDVQQHDA5T
    [...]
    -----END CERTIFICATE-----
  ssl_key: |
    -----BEGIN PRIVATE KEY-----
    MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQC5jdYbjtNTAKW4
    /CwQr/7wOiLGzVxChn3mmCIF3DwbL/qvTFTX2d8bDf6LjGwLYloXHscRfxszX/4h
    [...]
    -----END PRIVATE KEY-----
```

8.3.6 部署 NFS Ganesha

重要

NFS Ganesha 支持 NFS 4.1 和更高版本，不支持 NFS 3 版本。

cephadm 可使用预定义的 RADOS 存储池和可选的名称空间部署 NFS Ganesha。要部署 NFS Ganesha，请应用以下规范：

注意

您需要有一个预定义的 RADOS 存储池，否则 **`ceph orch apply`** 操作将失败。有关创建存储池的详细信息，请参见《管理和操作指南》, 第 18 章 “管理存储池”, 第 18.1 节 “创建存储池”。

```
service_type: nfs
```

```
service_id: EXAMPLE_NFS
placement:
  hosts:
  - ses-min1
  - ses-min2
spec:
  pool: EXAMPLE_POOL
  namespace: EXAMPLE_NAMESPACE
```

- EXAMPLE_NFS，包含用于标识 NFS 导出项的任意字符串。
- EXAMPLE_POOL，包含将存储 NFS Ganesha RADOS 配置对象的存储池的名称。
- EXAMPLE_NAMESPACE（可选），包含所需的对象网关 NFS 名称空间（例如，ganesha）。

8.3.7 部署 rbd-mirror

rbd-mirror 服务负责在两个 Ceph 集群之间同步 RADOS 块设备映像（有关更多详细信息，请参见《管理和操作指南》, 第 20 章 “RADOS 块设备”, 第 20.4 节 “RBD 映像镜像”）。要部署 rbd-mirror，请使用以下规范：

```
service_type: rbd-mirror
service_id: EXAMPLE_RBD_MIRROR
placement:
  hosts:
  - ses-min3
```

8.3.8 部署监控堆栈

监控堆栈包含 Prometheus、Prometheus 导出程序、Prometheus 告警管理器和 Grafana。Ceph Dashboard 使用这些组件来存储并直观呈现有关集群使用率和性能的详细度量。

提示

如果您的部署需要监控堆栈服务的自定义或本地提供的容器映像，请参见《管理和操作指南》, 第 16 章 “监控和告警”, 第 16.1 节 “配置自定义或本地映像”。

要部署监控堆栈，请执行以下步骤：

1. 在 Ceph Manager 守护进程中启用 `prometheus` 模块。这将公开内部 Ceph 度量，以便 Prometheus 可以读取这些信息：

```
cephuser@adm > ceph mgr module enable prometheus
```

注意

请确保在部署 Prometheus 之前运行此命令。如果部署前未运行该命令，则必须重新部署 Prometheus 才能更新 Prometheus 的配置：

```
cephuser@adm > ceph orch redeploy prometheus
```

2. 创建包含如下内容的规范文件（例如 `monitoring.yaml`）：

```
service_type: prometheus
placement:
  hosts:
  - ses-min2
---
service_type: node-exporter
---
service_type: alertmanager
placement:
  hosts:
  - ses-min4
---
service_type: grafana
placement:
  hosts:
```

```
  - ses-min3
```

3. 通过运行以下命令应用监控服务：

```
cephuser@adm > ceph orch apply -i monitoring.yaml
```

部署监控服务可能需要一到两分钟。

重要

Prometheus、Grafana 和 Ceph Dashboard 全都会自动配置为相互通讯，因而如上面所述进行部署时，Ceph Dashboard 中将实现功能完备的 Grafana 集成。

但使用 RBD 映像进行监控时不适用于此规则。有关更多信息，请参见《管理和操作指南》, 第 16 章 “监控和告警”, 第 16.5.4 节 “启用 RBD 映像监控”。

9 部署附加服务

9.1 安装 iSCSI 网关

iSCSI 是一种存储区域网络 (SAN) 协议，可让客户端（称作**发起程序**）将 SCSI 命令发送到远程服务器上的 SCSI 存储设备（**目标**）。SUSE Enterprise Storage 7.1 包含一个可通过 iSCSI 协议向异构客户端（例如 Microsoft Windows* 和 VMware* vSphere）开放 Ceph 存储管理的工具。多路径 iSCSI 访问可让这些客户端实现可用性与可伸缩性，此外，该标准化 iSCSI 协议在客户端与 SUSE Enterprise Storage 7.1 集群之间提供了一层额外的安全隔离。该配置工具名为 `ceph-iscsi`。使用 `ceph-iscsi`，Ceph 存储管理员可以定义精简配置的高可用性复制卷，该卷支持只读快照、读写克隆，以及 Ceph RADOS 块设备 (RBD) 的自动大小调整。然后，管理员可以通过单个 `ceph-iscsi` 网关主机或支持多路径故障转移的多个网关主机来导出卷。Linux、Microsoft Windows 和 VMware 主机可以使用 iSCSI 协议连接到卷，因此可像任何其他 SCSI 块设备一样供您使用。这意味着，SUSE Enterprise Storage 7.1 客户实际上可在 Ceph 上运行具有传统 SAN 所有功能和优势的完整块存储基础结构子系统，从而在未来实现蓬勃发展。

本章详细介绍如何设置 Ceph 集群基础结构和 iSCSI 网关，使客户端主机能够通过 iSCSI 协议，像在本地存储设备上一样使用远程存储的数据。

9.1.1 iSCSI 块存储

iSCSI 是 RFC 3720 中指定的、使用因特网协议 (IP) 的小型计算机系统接口 (SCSI) 命令集的一种实施。iSCSI 以服务形式实施，其中，客户端（发起程序）在 TCP 端口 3260 上通过会话来与服务器（目标）通讯。iSCSI 目标的 IP 地址和端口称为 **iSCSI 门户**，其中，一个目标可通过一个或多个端口公开。一个目标与一个或多个端口的组合称为**目标门户组** (TPG)。

iSCSI 的底层数据链路层协议通常为以太网。更具体地说，现代 iSCSI 基础结构使用 10 GigE 以太网或更快的网络实现最佳吞吐量。强烈建议在 iSCSI 网关与后端 Ceph 集群之间建立 10 Gb 以太网连接。

9.1.1.1 Linux 内核 iSCSI 目标

Linux 内核 iSCSI 目标最初称作 `linux-iscsi.or` 的 LIO，它是项目的原始域和网站。在过去一段时间，适用于 Linux 平台的 iSCSI 目标实施竞争产品不少于四个，但 LIO 作为单一 iSCSI 参照目标最终获得了压倒性优势。LIO 的主流内核代码使用简单但有点含糊的名称“目标”，旨在区分“目标核心”与各种前端和后端目标扩展模块。

可以说，最常用的前端扩展模块就是 iSCSI。但是，LIO 也支持光纤通道 (FC)、基于以太网的光纤通道 (FCoE) 和其他多种前端协议。目前，SUSE Enterprise Storage 仅支持 iSCSI 协议。

最常用的目标后端扩展模块是能够方便地在目标主机上重新导出任何可用块设备的扩展模块。此模块名为 **iblock**。但是，LIO 还有一个 RBD 特定的后端扩展模块，该扩展模块支持对 RBD 映像进行并行化多路径 I/O 访问。

9.1.1.2 iSCSI 发起程序

本节简要介绍 Linux、Microsoft Windows 和 VMware 平台上使用的 iSCSI 发起程序。

9.1.1.2.1 Linux

Linux 平台的标准发起程序是 `open-iscsi`。`open-iscsi` 会启动守护进程 `iscsid`，然后，用户可以使用该守护进程来发现任何给定端口上的 iSCSI 目标、登录到目标，以及映射 iSCSI 卷。`iscsid` 会与 SCSI 中间层通讯以创建内核中块设备，然后，内核便可像对待系统中任何其他的 SCSI 块设备一样来处理这些设备。可以结合设备映射程序多路径 (`open-iscsi`) 工具一起部署 `dm-multipath` 发起程序，以提供高度可用的 iSCSI 块设备。

9.1.1.2.2 Microsoft Windows 和 Hyper-V

Microsoft Windows 操作系统的默认 iSCSI 发起程序是 Microsoft iSCSI 发起程序。iSCSI 服务可通过图形用户界面 (GUI) 进行配置，并支持使用多路径 I/O 实现高可用性。

9.1.1.2.3 VMware

VMware vSphere 和 ESX 的默认 iSCSI 发起程序是 VMware ESX 软件 iSCSI 发起程序 `vmkiscsi`。启用该发起程序后，可通过 vSphere 客户端或使用 **`vmkiscsi-tool`** 命令对其进行配置。然后，可以使用 VMFS 来格式化通过 vSphere iSCSI 存储适配器连接的存储卷，并像使用任何其他 VM 存储设备一样使用它们。VMware 发起程序也支持使用多路径 I/O 实现高可用性。

9.1.2 关于 ceph-iscsi 的一般信息

`ceph-iscsi` 兼具 RADOS 块设备的优势与 iSCSI 无所不在的通用性。通过在 iSCSI 目标主机（称为 iSCSI 网关）上应用 `ceph-iscsi`，任何需要利用块存储的应用都可受益于 Ceph，即使不支持任何 Ceph 客户端协议的应用也不例外。而用户可以使用 iSCSI 或任何其他目标前端协议连接到 LIO 目标，从而可以转换针对 RBD 存储的所有目标 I/O。

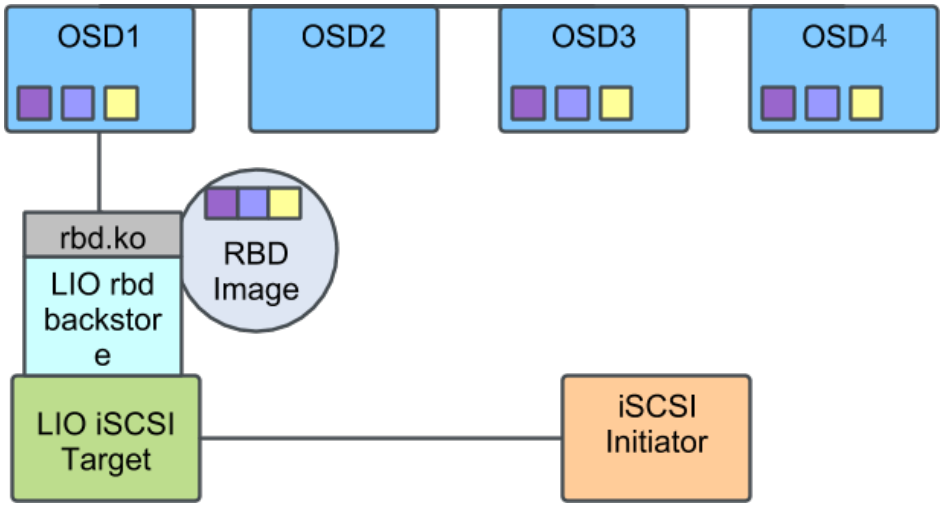

图 9.1： 包含单个 ISCSI 网关的 CEPH 集群

`ceph-iscsi` 先天就具有高可用性，并支持多路径操作。因此，下游发起程序主机可以使用多个 iSCSI 网关实现高可用性和可伸缩性。与包含多个网关的 iSCSI 配置通讯时，发起程序可在多个网关之间实现 iSCSI 请求的负载平衡。如果某个网关发生故障（暂时不可访问，或因为维护已被禁用），将通过另一个网关以透明方式继续处理 I/O。

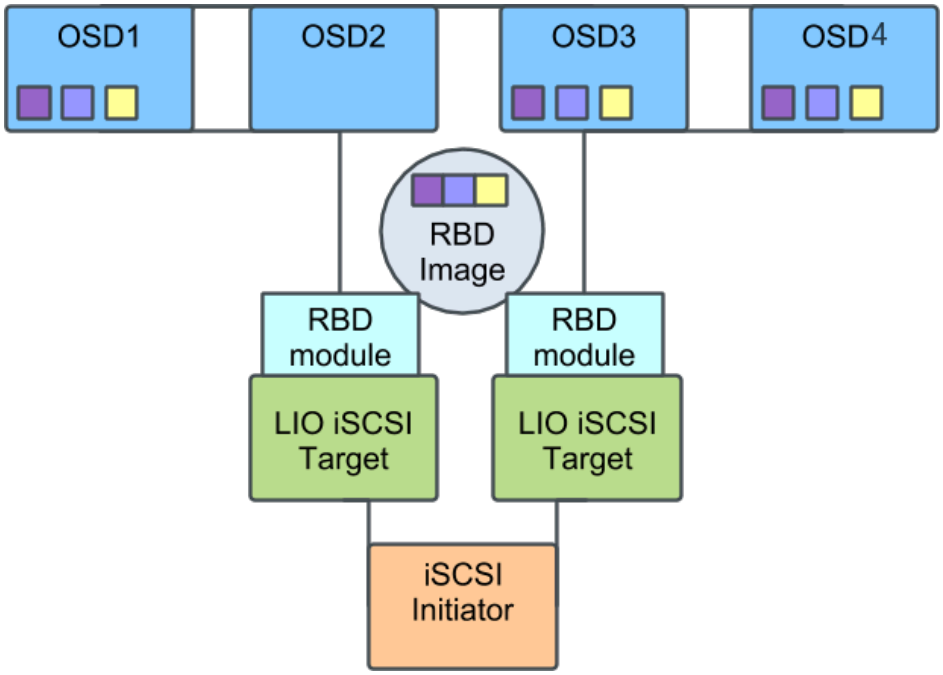

图 9.2： 包含多个 ISCSI 网关的 CEPH 集群

9.1.3 部署考虑事项

包含 `ceph-iscsi` 的最低 SUSE Enterprise Storage 7.1 配置由以下组件组成：

- 一个 Ceph 存储集群。该 Ceph 集群至少包括四台物理服务器，其中每台服务器至少托管八个对象存储守护进程 (OSD)。在此类配置中，有三个 OSD 节点额外充当 Monitor (MON) 主机。
- 一台通过 `ceph-iscsi` 配置且运行 LIO iSCSI 目标的 iSCSI 目标服务器。
- 一台 iSCSI 发起程序主机，它运行 `open-iscsi` (Linux)、Microsoft iSCSI 发起程序 (Microsoft Windows) 或任何其他兼容的 iSCSI 发起程序实施。

包含 `ceph-iscsi` 的建议 SUSE Enterprise Storage 7.1 生产配置由以下组件组成：

- 一个 Ceph 存储集群。一个 Ceph 生产集群由任意数量（通常是 10 个以上）的 OSD 节点（每个节点通常运行 10-12 个对象存储守护进程 (OSD)）以及至少三个专用 MON 主机组成。
- 多台通过 `ceph-iscsi` 配置且运行 LIO iSCSI 目标的 iSCSI 目标服务器。为实现 iSCSI 故障转移和负载平衡，这些服务器必须运行支持 `target_core_rbd` 模块的内核。可通过 SUSE Linux Enterprise Server 维护渠道获取更新软件包。
- 任意数量的 iSCSI 发起程序主机，这些主机运行 `open-iscsi` (Linux)、Microsoft iSCSI 发起程序 (Microsoft Windows) 或任何其他兼容的 iSCSI 发起程序实施。

9.1.4 安装和配置

本节介绍在 SUSE Enterprise Storage 的基础上安装和配置 iSCSI 网关的步骤。

9.1.4.1 将 iSCSI 网关部署到 Ceph 集群

Ceph iSCSI 网关采用与其他 Ceph 服务相同的过程进行部署，即使用 cephadm。有关细节，请参见第 8.3.5 节 “部署 iSCSI 网关”。

9.1.4.2 创建 RBD 映像

RBD 映像创建于 Ceph 存储区中，随后会导出到 iSCSI。建议为此使用专用的 RADOS 存储池。您可以在能使用 Ceph **rbd** 命令行实用程序连接到存储集群的任何主机上创建卷。这需要客户端至少有一个精简的 `ceph.conf` 配置文件，以及相应的 CephX 身份验证身份凭证。

要通过 iSCSI 创建一个随后可供导出的新卷，请使用 **rbd create** 命令并指定卷大小（以 MB 为单位）。例如，要在名为 `iscsi-images` 的存储池中创建名为 `testvol` 的 100 GB 卷，请运行以下命令：

```
cephuser@adm > rbd --pool iscsi-images create --size=102400 testvol
```

9.1.4.3 通过 iSCSI 导出 RBD 映像

要通过 iSCSI 导出 RBD 映像，您可以使用 Ceph Dashboard Web 界面或 `ceph-iscsi` gwcli 实用程序。在本节中，我们只重点介绍 gwcli，演示如何使用命令行创建导出 RBD 映像的 iSCSI 目标。

注意

无法通过 iSCSI 导出具有以下属性的 RBD 映像：

- 启用 `journaling` 功能的映像
- `stripe unit` 小于 4096 字节的映像

以 `root` 身份进入 iSCSI 网关容器：

```
# cephadm enter --name CONTAINER_NAME
```

以 `root` 身份启动 iSCSI 网关命令行界面：

```
# gwcli
```

转到 `iscsi-targets`，然后创建名为 `iqn.2003-01.org.linux-iscsi.iscsi.SYSTEM-ARCH:testvol` 的目标：

```
gwcli >  /> cd /iscsi-targets
gwcli >  /iscsi-targets> create iqn.2003-01.org.linux-iscsi.iscsi.SYSTEM-
ARCH:testvol
```

通过指定网关名称和 IP 地址创建 iSCSI 网关：

```
gwcli >  /iscsi-targets> cd iqn.2003-01.org.linux-iscsi.iscsi.SYSTEM-
ARCH:testvol/gateways
gwcli >  /iscsi-target...tvol/gateways> create iscsi1 192.168.124.104
gwcli >  /iscsi-target...tvol/gateways> create iscsi2 192.168.124.105
```

提示

使用 `help` 命令可显示当前配置节点中的可用命令列表。

在存储池 `iscsi-images` 中添加名为 `testvol` 的 RBD 映像：

```
gwcli >  /iscsi-target...tvol/gateways> cd /disks
gwcli >  /disks> attach iscsi-images/testvol
```

将 RBD 映像映射到目标：

```
gwcli >  /disks> cd /iscsi-targets/iqn.2003-01.org.linux-iscsi.iscsi.SYSTEM-
ARCH:testvol/disks
gwcli >  /iscsi-target...testvol/disks> add iscsi-images/testvol
```

注意

您可以使用级别较低的工具（例如 **`targetcli`**）来查询本地配置，但无法修改配置。

提示

您可以使用 **`ls`** 命令查看配置。有些配置节点还支持 **`info`** 命令，该命令可用于显示更多详细信息。

请注意，系统默认会启用 ACL 身份验证，因此目前尚不可访问此目标。有关身份验证和访问控制的详细信息，请参见第 9.1.4.4 节 “身份验证和访问控制”。

9.1.4.4 身份验证和访问控制

iSCSI 身份验证十分灵活，涵盖了许多身份验证可能性。

9.1.4.4.1 禁用 ACL 身份验证

无身份验证意味着任何发起程序均能访问相应目标上的任何 LUN。您可以通过禁用 ACL 身份验证来启用**无身份验证**：

```
gwcli >  /> cd /iscsi-targets/iqn.2003-01.org.linux-iscsi.iscsi.SYSTEM-
ARCH:testvol/hosts
gwcli >  /iscsi-target...testvol/hosts> auth disable_acl
```

9.1.4.4.2 使用 ACL 身份验证

使用基于发起程序名称的 ACL 身份验证时，只允许定义的发起程序进行连接。您可以通过运行以下命令来定义发起程序：

```
gwcli >  /> cd /iscsi-targets/iqn.2003-01.org.linux-iscsi.iscsi.SYSTEM-
ARCH:testvol/hosts
gwcli >  /iscsi-target...testvol/hosts> create
 iqn.1996-04.de.suse:01:e6ca28cc9f20
```

定义的发起程序虽然能够连接，但只能访问已明确添加到该发起程序的 RBD 映像：

```
gwcli >  /iscsi-target...:e6ca28cc9f20> disk add rbd/testvol
```

9.1.4.4.3 启用 CHAP 身份验证

除了 ACL 以外，您还可以通过为每个发起程序指定用户名和密码来启用 CHAP 身份验证：

```
gwcli >  /> cd /iscsi-targets/iqn.2003-01.org.linux-iscsi.iscsi.SYSTEM-
ARCH:testvol/hosts/iqn.1996-04.de.suse:01:e6ca28cc9f20
gwcli >  /iscsi-target...:e6ca28cc9f20> auth username=common12
 password=pass12345678
```

注意

用户名长度必须为 8 至 64 个字符，可以包含字母数字字符、`.`、`@`、`-`、`_` 或 `:`。

密码长度必须为 12 至 16 个字符，可以包含字母数字字符、`@`、`-`、`_` 或 `/`。

（可选）您也可以在 **`auth`** 命令中指定 `mutual_username` 和 `mutual_password` 参数，以启用 CHAP 相互身份验证。

9.1.4.4.4 配置发现和相互身份验证

发现身份验证独立于之前的身份验证方法。该身份验证需要身份凭证才能进行浏览，它是可选设置，可通过运行以下命令配置：

```
gwcli >  /> cd /iscsi-targets
gwcli >  /iscsi-targets> discovery_auth username=du123456 password=dp1234567890
```

注意

用户名长度必须为 8 至 64 个字符，只能包含字母、.、@、-、_ 或 :。

密码的长度必须为 12 至 16 个字符，并且只能包含字母、@、-、_ 或 /。

（可选）您也可以在 **discovery_auth** 命令中指定 mutual_username 和 mutual_password 参数。

可以使用以下命令来禁用发现身份验证：

```
gwcli >  /iscsi-targets> discovery_auth nochap
```

9.1.4.5 配置高级设置

可以为 ceph-iscsi 配置随后将传递给 LIO I/O 目标的高级参数。参数分为 target 和 disk 参数。

警告

除非另有说明，否则不建议将这些参数更改为使用非默认设置。

9.1.4.5.1 查看目标设置

您可以使用 **info** 命令查看这些设置的值：

```
gwcli >  /> cd /iscsi-targets/iqn.2003-01.org.linux-iscsi.iscsi.SYSTEM-
ARCH:testvol
gwcli >  /iscsi-target...i.SYSTEM-ARCH:testvol> info
```

还可以使用 **reconfigure** 命令更改设置：

```
gwcli >  /iscsi-target...i.SYSTEM-ARCH:testvol> reconfigure login_timeout 20
```

可用的 `target` 设置包括：

default_cmdsn_depth

默认的 CmdSN（命令顺序号）深度。限制 iSCSI 发起程序在任意时刻可拥有的未处理请求数量。

default_erl

默认的错误恢复级别。

login_timeout

登录超时值（以秒为单位）。

netif_timeout

NIC 故障超时（以秒为单位）。

prod_mode_write_protect

如果设置为 `1`，则阻止写入到 LUN。

9.1.4.5.2 查看磁盘设置

您可以使用 **`info`** 命令查看这些设置的值：

```
gwcli >  /> cd /disks/rbd/testvol
gwcli >  /disks/rbd/testvol> info
```

还可以使用 **`reconfigure`** 命令更改设置：

```
gwcli >  /disks/rbd/testvol> reconfigure rbd/testvol emulate_pr 0
```

可用的 `disk` 设置包括：

block_size

底层设备的块大小。

emulate_3pc

如果设置为 `1`，则启用“第三方复制”。

emulate_caw

如果设置为 `1`，则启用“比较并写入”。

emulate_dpo

如果设置为 1，则打开“禁用页面写出”。

emulate_fua_read

如果设置为 1，则启用“强制单元读取访问”。

emulate_fua_write

如果设置为 1，则启用“强制单元写入访问”。

emulate_model_alias

如果设置为 1，则使用后端设备名称作为模型别名。

emulate_pr

如果设置为 0，将禁用 SCSI 预留（包括永久组预留）支持。禁用该支持后，SES iSCSI 网关可能会忽略预留状态，导致请求延迟情况得到改进。

提示

如果 iSCSI 发起程序不需要 SCSI 预留支持，建议将 `backstore_emulate_pr` 设置为 `0`。

emulate_rest_reord

如果设置为 0，则队列算法修饰符的重新排序受限。

emulate_tas

如果设置为 1，则启用“任务已中止状态”。

emulate_tpu

如果设置为 1，则启用“精简配置 - 取消映射”。

emulate_tpws

如果设置为 1，则启用“精简配置 - 写入相同内容”。

emulate_ua_intlck_ctrl

如果设置为 1，则启用“单元警告联锁”。

emulate_write_cache

如果设置为 1，则打开“启用写入缓存”。

enforce_pr_isids

如果设置为 1，则强制永久性预留 ISID。

is_nonrot

如果设置为 1，则后备存储为非旋转设备。

max_unmap_block_desc_count

UNMAP 的最大块描述符数。

max_unmap_lba_count：

UNMAP 的最大 LBA 数。

max_write_same_len

WRITE_SAME 的最大长度。

optimal_sectors

扇区中的最佳请求大小。

pi_prot_type

DIF 保护类型。

queue_depth

队列深度。

unmap_granularity

UNMAP 粒度。

unmap_granularity_alignment

UNMAP 粒度对齐。

force_pr_aptpl

如果启用该设置，无论客户端是否通过 **aptpl=1** 发出了相应请求，LIO 都始终会将永久预留状态写出到永久存储区。这对 LIO 的内核 RBD 后端不会产生任何影响，该后端始终会保留 PR 状态。理论上，如果有人尝试通过配置禁用该设置，`target_core_rbd` 选项应该会强制将其设置为“1”并抛出错误。

unmap_zeroes_data

影响 LIO 是否会向 SCSI 发起程序公布 LBPRZ，表示将在执行包含取消映射位的 UNMAP 或 WRITE SAME 命令后从某个区域回读零。

9.1.5 使用 tcmu-runner 导出 RADOS 块设备映像

ceph-iscsi 支持 rbd（基于内核）和 user:rbd (tcmu-runner) 后备存储，这使整个管理过程变得透明，并且独立于后备存储。

警告：技术预览

基于 tcmu-runner 的 iSCSI 网关部署目前以技术预览的方式提供。

与基于内核的 iSCSI 网关部署不同，基于 tcmu-runner 的 iSCSI 网关不支持多路径 I/O 或 SCSI 永久性预留。

要使用 tcmu-runner 导出 RADOS 块设备映像，您只需在挂接磁盘时指定 user:rbd 后备存储即可：

```
gwcli >  /disks> attach rbd/testvol backstore=user:rbd
```

注意

使用 tcmu-runner 时，导出的 RBD 映像必须启用 exclusive-lock 特性。

III 从旧版本升级

10 从 SUSE Enterprise Storage 6 升级到版本 7.1 87

11 从 SUSE Enterprise Storage 7 升级到版本 7.1 111

10 从 SUSE Enterprise Storage 6 升级到版本 7.1

本章说明将 SUSE Enterprise Storage 6 升级到版本 7.1 的步骤。

升级包括以下任务：

- 从 Ceph Nautilus 升级到 Pacific。
- 从通过 RPM 软件包安装和运行 Ceph 切换到在容器中运行。
- 完全删除 DeepSea 并以 `ceph-salt` 和 cephadm 替代。

警告

本章中的升级信息**仅**适用于从 DeepSea 到 cephadm 的升级。如果您要在 SUSE CaaS 平台上部署 SUSE Enterprise Storage，请不要尝试按照这些说明操作。

重要

不支持从低于 6 的 SUSE Enterprise Storage 版本升级。您需要先升级到 SUSE Enterprise Storage 6 的最新版本，然后再按照本章中所述的步骤操作。

10.1 升级前

开始升级之前，**必须**完成以下任务。在 SUSE Enterprise Storage 6 生命周期内可随时执行这些任务。

- **必须**在升级前执行从 FileStore 到 BlueStore 的 OSD 迁移，因为 FileStore 在 SUSE Enterprise Storage 7.1 中不受支持。有关 BlueStore 以及如何从 FileStore 迁移的更多详细信息，请参见 https://documentation.suse.com/ses/6/single-html/ses-deployment/#filestore2bluestore。
- 如果您运行的是仍在使用 `ceph-disk` OSD 的旧集群，则**需要**在升级前切换到 `ceph-volume`。有关详细信息，请参见https://documentation.suse.com/ses/6/single-html/ses-deployment/#upgrade-osd-deployment。

10.1.1 需考虑的要点

升级前，请务必通读以下内容，确保您了解所有需要执行的任务。

- **阅读发行说明**。在发行说明中，可以找到更多有关自 SUSE Enterprise Storage 的上一个版本发行后所进行的更改的信息。检查发行说明以了解：
  - 您的硬件是否有特殊注意事项。
  - 所使用的任何软件包是否发生了重大更改。
  - 是否需要对您的安装实施特殊预防措施。

  发行说明还提供未能及时编入手册中的信息。它们还包含有关已知问题的说明。
  您可以在 https://www.suse.com/releasenotes/ 上找到联机 SES 7.1 发行说明。
  此外，安装 SES 7.1 软件源中的 `release-notes-ses` 软件包之后，可在本地的 `/usr/share/doc/release-notes` 目录中或 https://www.suse.com/releasenotes/ 网页上找到发行说明。
- 阅读第 II 部分 “部署 Ceph 集群”，熟悉 `ceph-salt` 和 Ceph orchestrator，尤其要了解有关服务规范的信息。
- 升级集群可能需要花很长时间 - 所需时间大约为升级一台计算机的时间乘以集群节点数。
- 您需要先升级 Salt 主控端，然后将 DeepSea 替换为 `ceph-salt` 和 cephadm。至少要等到所有 Ceph Manager 节点都升级后，您**才能**开始使用 cephadm orchestrator 模块。
- 从使用 Nautilus RPM 升级到使用 Pacific 容器需要一步到位。这意味着您需要一次升级整个节点，而不是一次仅升级一个守护进程。
- 核心服务（MON、MGR、OSD）的升级是有序进行的。每个服务在升级期间都可用。升级核心服务后需要重新部署网关服务（元数据服务器、对象网关、NFS Ganesha、iSCSI 网关）。下面每个服务都有特定的停机时间：

  - **重要**

    元数据服务器和对象网关的停机时间自节点从 SUSE Linux Enterprise Server 15 SP1 升级到 SUSE Linux Enterprise Server 15 SP3 开始，直到升级过程结束时重新部署这些服务为止。特别是在这些服务与 MON、MGR 或 OSD 共置

时更要考虑到这一点，因为它们可能会在整个集群升级过程中都处于停用状态。如果这对您是个问题，请考虑在升级前在另外的节点上分别部署这些服务，以尽可能缩短它们处于停用状态的时间。如此，停机时间便将是网关节点升级的持续时间，而不是整个集群升级的持续时间。

- 从 SUSE Linux Enterprise Server 15 SP1 升级到 SUSE Linux Enterprise Server 15 SP3 期间，NFS Ganesha 和 iSCSI 网关仅会在节点重引导时处于停用状态，并会在以容器化模式重新部署每个服务时再次短暂处于停用状态。

10.1.2 备份集群配置和数据

强烈建议您在开始升级到 SUSE Enterprise Storage 7.1 之前备份所有集群配置和数据。有关如何备份所有数据的说明，请参见 https://documentation.suse.com/ses/6/single-html/ses-admin/#cha-deployment-backup 。

10.1.3 确认先前升级的步骤

如果您先前是从版本 5 升级的，请确认升级到版本 6 的过程已成功完成：

检查 `/srv/salt/ceph/configuration/files/ceph.conf.import` 文件是否存在。

此文件是在从 SUSE Enterprise Storage 5 升级到 6 期间由 engulf 进程创建的。`configuration_init: default-import` 选项在 `/srv/pillar/ceph/proposals/config/stack/default/ceph/cluster.yml` 中设置。

如果 `configuration_init` 仍设为 `default-import`，则表示集群使用 `ceph.conf.import` 而非 DeepSea 的默认 `ceph.conf` 作为其配置文件，后者基于 `/srv/salt/ceph/configuration/files/ceph.conf.d/` 中的文件进行编译。

因此，您需要检查 `ceph.conf.import` 中是否有任何自定义配置，并根据需要将相应配置移到 `/srv/salt/ceph/configuration/files/ceph.conf.d/` 中的其中一个文件中。

然后从 `/srv/pillar/ceph/proposals/config/stack/default/ceph/cluster.yml` 中删除 `configuration_init: default-import` 行。

10.1.4 更新集群节点并确认集群健康状况

确认 SUSE Linux Enterprise Server 15 SP1和 SUSE Enterprise Storage 6 的所有最新更新是否已应用到所有集群节点：

```
# zypper refresh && zypper patch
```

提示

有关更新集群节点的详细信息，请参见 https://documentation.suse.com/ses/6/html/ses-all/storage-salt-cluster.html#deepsea-rolling-updates。

应用更新后，重启动 Salt 主控端，同步新 Salt 模块，然后检查集群健康状况：

```
root@master # systemctl restart salt-master.service
root@master # salt '*' saltutil.sync_all
cephuser@adm > ceph -s
```

10.1.4.1 禁用不安全的客户端

从 Nautilus v14.2.20 起，引入了新的健康状况警告来告知您允许了不安全的客户端加入集群。此警告默认会**打开**。Ceph Dashboard 会表明集群处于 HEALTH_WARN 状态。命令行会按如下所示校验集群状态：

```
cephuser@adm > ceph status
cluster:
  id:     3fe8b35a-689f-4970-819d-0e6b11f6707c
  health: HEALTH_WARN
  mons are allowing insecure global_id reclaim
[...]
```

此警告表示 Ceph Monitor 仍在允许未增补的旧版客户端连接集群。这样可确保在升级集群时，现有客户端仍可连接集群，但会警告您存在需要解决的问题。当集群和所有客户端都升级到最新版本的 Ceph 后，运行以下命令禁用未增补的客户端：

```
cephuser@adm > ceph config set mon auth_allow_insecure_global_id_reclaim false
```

10.1.5 确认对软件源和容器映像的访问权限

确认是否每个集群节点都能访问 SUSE Linux Enterprise Server 15 SP3 和 SUSE Enterprise Storage 7.1 软件源以及容器映像注册表。

10.1.5.1 软件源

如果所有节点都已在 SCC 中注册，您便可以使用 **`zypper migration`** 命令进行升级。有关更多详细信息，请参见 https://documentation.suse.com/sles/15-SP3/html/SLES-all/cha-upgrade-online.html#sec-upgrade-online-zypper。

如果节点**未**在 SCC 中注册，请禁用所有现有软件源，并为以下每个扩展添加 `Pool` 和 `Updates` 软件源：

- SLE-Product-SLES/15-SP3
- SLE-Module-Basesystem/15-SP3
- SLE-Module-Server-Applications/15-SP3
- SUSE-Enterprise-Storage-7.1

10.1.5.2 容器映像

所有集群节点都需要访问容器映像注册表。在大多数情况下，您将使用 `registry.suse.com` 上的公共 SUSE 注册表。您需要以下映像：

- registry.suse.com/ses/7.1/ceph/ceph
- registry.suse.com/ses/7.1/ceph/grafana
- registry.suse.com/ses/7.1/ceph/prometheus-server
- registry.suse.com/ses/7.1/ceph/prometheus-node-exporter
- registry.suse.com/ses/7.1/ceph/prometheus-alertmanager

或者，例如对于实体隔离部署，请配置本地注册表并确认您是否有一组正确的容器映像可用。有关配置本地容器映像注册表的更多详细信息，请参见第 7.2.10 节 “使用容器注册表”。

10.2 升级 Salt 主控端

以下过程描述了升级 Salt 主控端的过程：

1. 将底层操作系统升级到 SUSE Linux Enterprise Server 15 SP3：

   - 对于所有节点都已在 SCC 中注册的集群，请运行 **`zypper migration`**。
   - 对于节点具有手动指定软件源的集群，请运行 **`zypper dup`**，然后再运行 **`reboot`**。

2. 禁用 DeepSea 阶段以避免意外使用。将以下内容添加到 `/srv/pillar/ceph/stack/global.yml`：

   ```
   stage_prep: disabled
   stage_discovery: disabled
   stage_configure: disabled
   stage_deploy: disabled
   stage_services: disabled
   stage_remove: disabled
   ```

   保存文件并应用更改：

   ```
   root@master # salt '*' saltutil.pillar_refresh
   ```

3. 如果您**未**使用 `registry.suse.com` 中的容器映像，而是使用本地配置的注册表，请编辑 `/srv/pillar/ceph/stack/global.yml` 以告知 DeepSea 要使用哪个 Ceph 容器映像和注册表。例如，要使用 `192.168.121.1:5000/my/ceph/image`，请添加下面几行：

   ```
   ses7_container_image: 192.168.121.1:5000/my/ceph/image
   ses7_container_registries:
     - location: 192.168.121.1:5000
   ```

   如果您需要为注册表指定身份验证信息，请添加 `ses7_container_registry_auth:` 块，例如：

   ```
   ses7_container_image: 192.168.121.1:5000/my/ceph/image
   ```

```
ses7_container_registries:
  - location: 192.168.121.1:5000
ses7_container_registry_auth:
  registry: 192.168.121.1:5000
  username: USER_NAME
  password: PASSWORD
```

保存文件并应用更改：

```
root@master # salt '*' saltutil.refresh_pillar
```

4. 同化现有配置：

```
cephuser@adm > ceph config assimilate-conf -i /etc/ceph/ceph.conf
```

5. 确认升级状态。您的输出可能因集群配置而异：

```
root@master # salt-run upgrade.status
The newest installed software versions are:
 ceph: ceph version 16.2.7-640-gceb23c7491b
 (ceb23c7491bd96ab7956111374219a4cdcf6f8f4) pacific (stable)
 os: SUSE Linux Enterprise Server 15 SP3

Nodes running these software versions:
 admin.ceph (assigned roles: master, prometheus, grafana)

Nodes running older software versions must be upgraded in the following
 order:
 1: mon1.ceph (assigned roles: admin, mon, mgr)
 2: mon2.ceph (assigned roles: admin, mon, mgr)
 3: mon3.ceph (assigned roles: admin, mon, mgr)
 4: data4.ceph (assigned roles: storage, mds)
 5: data1.ceph (assigned roles: storage)
 6: data2.ceph (assigned roles: storage)
 7: data3.ceph (assigned roles: storage)
 8: data5.ceph (assigned roles: storage, rgw)
```

10.3 升级 MON、MGR 和 OSD 节点

每次升级一个 Ceph Monitor、Ceph Manager 和 OSD 节点。对于每个服务，请执行以下步骤：

1. 采用任何 OSD 节点之前，需要执行 OSD 节点的格式转换以改进 OMAP 数据的核算。您可以通过在管理节点上运行以下命令来执行此操作：

```
cephuser@adm > ceph config set osd bluestore_fsck_quick_fix_on_mount true
```

OSD 节点将会在采用过程完成后自动进行转换。

注意

转换所需的时间可能从数分钟到数小时不等，具体时长取决于相关硬盘包含的 OMAP 数据量。有关详细信息，请参考 https://docs.ceph.com/en/latest/releases/pacific/#upgrading-non-cephadm-clusters 。

2. 如果您要升级的是 OSD 节点，请通过运行以下命令避免在升级过程中将 OSD 标记为 `out`：

```
cephuser@adm > ceph osd add-noout SHORT_NODE_NAME
```

将 `SHORT_NODE_NAME` 替换为 **`ceph osd tree`** 命令输出中显示的节点的简短名称。在以下输入中，主机简短名称为 `ses-min1` 和 `ses-min2`

```
root@master # ceph osd tree
ID   CLASS  WEIGHT   TYPE NAME       STATUS  REWEIGHT  PRI-AFF
 -1         0.60405  root default
-11         0.11691      host ses-min1
  4    hdd  0.01949          osd.4       up   1.00000  1.00000
  9    hdd  0.01949          osd.9       up   1.00000  1.00000
 13    hdd  0.01949          osd.13      up   1.00000  1.00000
[...]
 -5         0.11691      host ses-min2
  2    hdd  0.01949          osd.2       up   1.00000  1.00000
  5    hdd  0.01949          osd.5       up   1.00000  1.00000
[...]
```

3. 将底层操作系统升级到 SUSE Linux Enterprise Server 15 SP3：

   - 如果集群的节点都已在 SCC 中注册，请运行 **`zypper migration`**。
   - 如果集群的节点具有手动指定的软件源，请运行 **`zypper dup`**，然后再运行 **`reboot`**。

4. 重引导节点后，通过在 Salt 主控端上运行以下命令，将该节点上所有现有 MON、MGR 和 OSD 守护进程进行容器化：

   ```
   root@master # salt MINION_ID state.apply ceph.upgrade.ses7.adopt
   ```

   将 `MINION_ID` 替换为您要升级的受控端的 ID。您可以通过在 Salt 主控端上运行 **`salt-key -L`** 命令来获取受控端 ID 的列表。

   提示

   要查看**采用**的状态和进度，请检查 Ceph Dashboard 或在 Salt 主控端上运行下面其中一个命令：

   ```
   root@master # ceph status
   root@master # ceph versions
   root@master # salt-run upgrade.status
   ```

5. 成功完成采用后，如果您要升级的节点是 OSD 节点，请取消设置 `noout` 标志：

   ```
   cephuser@adm > ceph osd rm-noout SHORT_NODE_NAME
   ```

10.4 升级网关节点

接下来，需升级各个网关节点（Samba 网关、元数据服务器、对象网关、NFS Ganesha 或 iSCSI 网关）。针对每个节点将底层操作系统升级到 SUSE Linux Enterprise Server 15 SP3：

- 如果集群的节点都已在 SUSE Customer Center 中注册，请运行 **zypper migration** 命令。
- 如果集群的节点具有手动指定的软件源，请运行 **zypper dup**，然后再运行 **reboot** 命令。

此步骤也适用于属于集群但尚未指定任何角色的任何节点（如有疑问，请查看通过 **salt-key -L** 命令提供的 Salt 主控端上的主机列表，并将其与 **salt-run upgrade.status** 命令的输出进行比较）。

升级集群中所有节点上的操作系统后，下一步是安装 `ceph-salt` 软件包并应用集群配置。升级过程结束时，会以容器化模式重新部署实际的网关服务。

注意

从升级到 SUSE Linux Enterprise Server 15 SP3 开始，直至升级过程结束时重新部署元数据服务器和对象网关服务为止，这段期间这些服务将不可用。

10.5 安装 ceph-salt 并应用集群配置

在开始安装 `ceph-salt` 并应用集群配置这一过程前，请先通过运行以下命令查看集群和升级状态：

```
root@master # ceph status
root@master # ceph versions
root@master # salt-run upgrade.status
```

1. 删除 DeepSea 创建的 `rbd_exporter` 和 `rgw_exporter` 定时任务 (cron job)。在 Salt 主控端上，以 `root` 身份运行 **crontab -e** 命令以编辑 crontab。删除下列项目（如果存在）：

```
# SALT_CRON_IDENTIFIER:deepsea rbd_exporter cron job
*/5 * * * * /var/lib/prometheus/node-exporter/rbd.sh > \
 /var/lib/prometheus/node-exporter/rbd.prom 2> /dev/null
# SALT_CRON_IDENTIFIER:Prometheus rgw_exporter cron job
```

```
*/5 * * * * /var/lib/prometheus/node-exporter/ceph_rgw.py > \
 /var/lib/prometheus/node-exporter/ceph_rgw.prom 2> /dev/null
```

2. 通过运行以下命令从 DeepSea 导出集群配置：

```
root@master # salt-run upgrade.ceph_salt_config > ceph-salt-config.json
root@master # salt-run upgrade.generate_service_specs > specs.yaml
```

3. 卸装 DeepSea，并在 Salt 主控端上安装 `ceph-salt`：

```
root@master # zypper remove 'deepsea*'
root@master # zypper install ceph-salt
```

4. 重启动 Salt 主控端并同步 Salt 模块：

```
root@master # systemctl restart salt-master.service
root@master # salt \* saltutil.sync_all
```

5. 将 DeepSea 的集群配置导入 `ceph-salt`：

```
root@master # ceph-salt import ceph-salt-config.json
```

6. 为集群节点通讯生成 SSH 密钥：

```
root@master # ceph-salt config /ssh generate
```

提示

确认是否已从 DeepSea 导入集群配置，并指定可能缺失的选项：

```
root@master # ceph-salt config ls
```

有关集群配置的完整描述，请参见第 7.2 节 “配置集群属性”。

7. 应用配置并启用 cephadm：

```
root@master # ceph-salt apply
```

8. 如果您需要提供本地容器注册表 URL 和访问身份凭证，请按照第 7.2.10 节 “使用容器注册表”中所述的步骤操作。

9. 如果您使用的**不是** `registry.suse.com` 中的容器映像，而是本地配置的注册表，请运行以下命令告知 Ceph 要使用的容器映像

```
root@master # ceph config set global container_image IMAGE_NAME
```

例如：

```
root@master # ceph config set global container_image 192.168.121.1:5000/my/ceph/image
```

10. 停止并禁用 SUSE Enterprise Storage 6 `ceph-crash` 守护进程。将于稍后自动启动这些守护进程的新容器化格式。

```
root@master # salt '*' service.stop ceph-crash
root@master # salt '*' service.disable ceph-crash
```

10.6 升级并采用监控堆栈

以下过程会采用监控堆栈的所有组件（有关更多详细信息，请参见《管理和操作指南》，第 16 章 “监控和告警”）。

1. 暂停 orchestrator：

```
cephuser@adm > ceph orch pause
```

2. 在运行 Prometheus、Grafana 和告警管理器（默认为 Salt 主控端）的节点上，运行以下命令：

```
cephuser@adm > cephadm adopt --style=legacy --name prometheus.$(hostname)
cephuser@adm > cephadm adopt --style=legacy --name alertmanager.$(hostname)
cephuser@adm > cephadm adopt --style=legacy --name grafana.$(hostname)
```

提示

如果您运行的**不是**默认容器映像注册表 `registry.suse.com`，则需要在每个命令中指定要使用的映像，例如：

```
cephuser@adm > cephadm --image 192.168.121.1:5000/ses/7.1/ceph/
prometheus-server:2.32.1 \
  adopt --style=legacy --name prometheus.$(hostname)
cephuser@adm > cephadm --image 192.168.121.1:5000/ses/7.1/ceph/
prometheus-alertmanager:0.21.0 \
  adopt --style=legacy --name alertmanager.$(hostname)
cephuser@adm > cephadm --image 192.168.121.1:5000/ses/7.1/ceph/
grafana:7.5.12 \
 adopt --style=legacy --name grafana.$(hostname)
```

《管理和操作指南》, 第 16 章 “监控和告警”, 第 16.1 节 “配置自定义或本地映像”中列出了需要的容器映像及相应版本。

3. 从**所有**节点中删除 Node-ExporterNode-Exporter 不需要迁移，将在应用 `specs.yaml` 文件时作为容器重新安装。

```
> sudo zypper rm golang-github-prometheus-node_exporter
```

或者，您可以在管理节点上使用 Salt 同时从所有节点中删除 Node-Exporter：

```
root@master # salt '*' pkg.remove golang-github-prometheus-node_exporter
```

4. 应用您之前从 DeepSea 导出的服务规范：

```
cephuser@adm > ceph orch apply -i specs.yaml
```

提示

如果您运行的**不是**默认容器映像注册表 `registry.suse.com`，而是本地容器注册表，请在部署 Node-Exporter 前对 cephadm 进行配置，以便为 Node-Exporter 部署使用本地注册表中的容器映像。否则，您可以放心跳过此步骤并忽略以下警告：

```
cephuser@adm > ceph config set mgr mgr/cephadm/
container_image_node_exporter QUALIFIED_IMAGE_PATH
```

确保监控服务的所有容器映像都指向本地注册表，而不仅仅是 Node-Exporter 的映像。此步骤只要求您对 Node-Exporter 执行此操作，但建议您此时在 cephadm 中将所有监控容器映像都设置为指向本地注册表。

如果您未这样做，监控服务的新部署以及重新部署过程将使用默认的 cephadm 配置，您最终可能无法部署服务（如果是实体隔离部署），或者最终会部署混合版本的服务。

《管理和操作指南》, 第 16 章 “监控和告警”, 第 16.1 节 “配置自定义或本地映像”中介绍了需如何配置 cephadm 以使用本地注册表中的容器映像。

5. 继续 orchestrator：

```
cephuser@adm > ceph orch resume
```

10.7 重新部署网关服务

10.7.1 升级对象网关

在 SUSE Enterprise Storage 7.1 中，对象网关始终配置有一个领域，以为未来使用多站点提供支持（有关更多详细信息，请参见《管理和操作指南》, 第 21 章 “Ceph 对象网关”, 第 21.13 节 “多站点对象网关”）。如果您在 SUSE Enterprise Storage 6 中使用了单站点对象网关配置，请按照以下步骤添加领域。如果您不打算使用多站点功能，则可以为领域、区域组和区域名称使用 `default` 值。

1. 创建新领域：

```
cephuser@adm > radosgw-admin realm create --rgw-realm=REALM_NAME --default
```

2. （可选）重命名默认区域和区域组。

```
cephuser@adm > radosgw-admin zonegroup rename \
 --rgw-zonegroup default \
 --zonegroup-new-name=ZONEGROUP_NAME
cephuser@adm > radosgw-admin zone rename \
```

```
 --rgw-zone default \
 --zone-new-name ZONE_NAME \
 --rgw-zonegroup=ZONEGROUP_NAME
```

3. 配置主区域组：

```
cephuser@adm > radosgw-admin zonegroup modify \
 --rgw-realm=REALM_NAME \
 --rgw-zonegroup=ZONEGROUP_NAME \
 --endpoints http://RGW.EXAMPLE.COM:80 \
 --master --default
```

4. 配置主区域。为此，您将需要启用了 `system` 标志的对象网关用户的 ACCESS_KEY 和 SECRET_KEY。这通常是 `admin` 用户。要获取 ACCESS_KEY 和 SECRET_KEY，请运行 **`radosgw-admin user info --uid admin --rgw-zone=`**ZONE_NAME。

```
cephuser@adm > radosgw-admin zone modify \
 --rgw-realm=REALM_NAME \
 --rgw-zonegroup=ZONEGROUP_NAME \
 --rgw-zone=ZONE_NAME \
 --endpoints http://RGW.EXAMPLE.COM:80 \
 --access-key=ACCESS_KEY \
 --secret=SECRET_KEY \
 --master --default
```

5. 提交更新后的配置：

```
cephuser@adm > radosgw-admin period update --commit
```

要将对象网关服务容器化，请按第 8.3.4 节 “部署对象网关”中所述创建其规范文件，然后应用该文件。

```
cephuser@adm > ceph orch apply -i RGW.yml
```

10.7.2 升级 NFS Ganesha

重要

NFS Ganesha 支持 NFS 4.1 和更高版本，不支持 NFS 3 版本。

下面演示如何将运行 Ceph Nautilus 的现有 NFS Ganesha 服务迁移到运行 Ceph Octopus 的 NFS Ganesha 容器。

警告

以下文档要求您已成功升级了核心 Ceph 服务。

NFS Ganesha 会在 RADOS 存储池中存储额外的守护进程特定配置并导出配置。所配置的 RADOS 存储池可在 `ganesha.conf` 文件的 `RADOS_URLS` 块的 `watch_url` 行上找到。默认情况下，此存储池将重命名为 `ganesha_config`

强烈建议您在尝试进行任何迁移之前针对导出和位于 RADOS 存储池中的守护进程配置对象创建一份副本。要找到配置的 RADOS 存储池，请运行以下命令：

```
cephuser@adm > grep -A5 RADOS_URLS /etc/ganesha/ganesha.conf
```

列出 RADOS 存储池的内容：

```
cephuser@adm > rados --pool ganesha_config --namespace ganesha ls | sort
  conf-node3
  export-1
  export-2
  export-3
  export-4
```

复制 RADOS 对象：

```
cephuser@adm > RADOS_ARGS="--pool ganesha_config --namespace ganesha"
cephuser@adm > OBJS=$(rados $RADOS_ARGS ls)
cephuser@adm > for obj in $OBJS; do rados $RADOS_ARGS get $obj $obj; done
cephuser@adm > ls -lah
```

```
total 40K
drwxr-xr-x 2 root root 4.0K Sep 8 03:30 .
drwx------ 9 root root 4.0K Sep 8 03:23 ..
-rw-r--r-- 1 root root 90 Sep 8 03:30 conf-node2
-rw-r--r-- 1 root root 90 Sep 8 03:30 conf-node3
-rw-r--r-- 1 root root 350 Sep 8 03:30 export-1
-rw-r--r-- 1 root root 350 Sep 8 03:30 export-2
-rw-r--r-- 1 root root 350 Sep 8 03:30 export-3
-rw-r--r-- 1 root root 358 Sep 8 03:30 export-4
```

对于每个节点，需要停止现有的 NFS Ganesha 服务，然后将其替换为由 cephadm 管理的容器。

1. 停止并禁用现有的 NFS Ganesha 服务：

   ```
   cephuser@adm > systemctl stop nfs-ganesha
   cephuser@adm > systemctl disable nfs-ganesha
   ```

2. 停止现有的 NFS Ganesha 服务之后，可以使用 cephadm 在容器中部署一个新的服务。为此，您需要创建一个服务规范，其中包含将用于标识此新 NFS 集群的 `service_id`、在归置规范中作为主机列出的所要迁移节点的主机名，以及包含所配置 NFS 导出对象的 RADOS 存储池和名称空间。例如：

   ```
   service_type: nfs
   service_id: SERVICE_ID
   placement:
     hosts:
     - node2
     pool: ganesha_config
     namespace: ganesha
   ```

   有关创建归置规范的详细信息，请参见第 8.2 节 “服务和归置规范”。

3. 应用归置规范：

   ```
   cephuser@adm > ceph orch apply -i FILENAME.yaml
   ```

4. 确认主机上是否运行有 NFS Ganesha 守护进程：

```
cephuser@adm > ceph orch ps --daemon_type nfs
NAME           HOST   STATUS         REFRESHED  AGE  VERSION  IMAGE NAME
                              IMAGE ID      CONTAINER ID
nfs.foo.node2  node2  running (26m)  8m ago     27m  3.3
 registry.suse.com/ses/7.1/ceph/ceph:latest  8b4be7c42abd  c8b75d7c8f0d
```

5. 针对每个 NFS Ganesha 节点重复上述步骤。您不需要为每个节点创建单独的服务规范。您只需将每个节点的主机名添加到现有 NFS 服务规范中，然后重新应用该规范。

现有的导出可以通过两种方法进行迁移：

- 使用 Ceph Dashboard 手动重新创建或重新指定。
- 手动将每个守护进程 RADOS 对象的内容复制到新创建的 NFS Ganesha 通用配置中。

过程 10.1： 手动将导出复制到 NFS GANESHA 通用配置文件

1. 确定每个守护进程 RADOS 对象的列表：

```
cephuser@adm > RADOS_ARGS="--pool ganesha_config --namespace ganesha"
cephuser@adm > DAEMON_OBJS=$(rados $RADOS_ARGS ls | grep 'conf-')
```

2. 复制每个守护进程的 RADOS 对象：

```
cephuser@adm > for obj in $DAEMON_OBJS; do rados $RADOS_ARGS get $obj $obj;
 done
cephuser@adm > ls -lah
total 20K
drwxr-xr-x 2 root root 4.0K Sep 8 16:51 .
drwxr-xr-x 3 root root 4.0K Sep 8 16:47 ..
-rw-r--r-- 1 root root 90 Sep 8 16:51 conf-nfs.SERVICE_ID
-rw-r--r-- 1 root root 90 Sep 8 16:51 conf-node2
-rw-r--r-- 1 root root 90 Sep 8 16:51 conf-node3
```

3. 排序并合并到单个导出列表中：

```
cephuser@adm > cat conf-* | sort -u > conf-nfs.SERVICE_ID
cephuser@adm > cat conf-nfs.foo
%url "rados://ganesha_config/ganesha/export-1"
%url "rados://ganesha_config/ganesha/export-2"
```

```
%url "rados://ganesha_config/ganesha/export-3"
%url "rados://ganesha_config/ganesha/export-4"
```

4. 写入新的 NFS Ganesha 通用配置文件：

```
cephuser@adm > rados $RADOS_ARGS put conf-nfs.SERVICE_ID conf-
nfs.SERVICE_ID
```

5. 通知 NFS Ganesha 守护进程：

```
cephuser@adm > rados $RADOS_ARGS notify conf-nfs.SERVICE_ID conf-
nfs.SERVICE_ID
```

注意

此操作将导致守护进程重新加载配置。

成功迁移服务后，可以删除基于 Nautilus 的 NFS Ganesha 服务。

1. 删除 NFS Ganesha：

```
cephuser@adm > zypper rm nfs-ganesha
Reading installed packages...
Resolving package dependencies...
The following 5 packages are going to be REMOVED:
  nfs-ganesha nfs-ganesha-ceph nfs-ganesha-rados-grace nfs-ganesha-rados-
urls nfs-ganesha-rgw
5 packages to remove.
After the operation, 308.9 KiB will be freed.
Continue? [y/n/v/...? shows all options] (y): y
(1/5) Removing nfs-ganesha-
ceph-2.8.3+git0.d504d374e-3.3.1.x86_64 ..........................................
[done]
(2/5) Removing nfs-ganesha-
rgw-2.8.3+git0.d504d374e-3.3.1.x86_64 ...........................................
[done]
(3/5) Removing nfs-ganesha-rados-
urls-2.8.3+git0.d504d374e-3.3.1.x86_64 ..........................................
[done]
```

```
(4/5) Removing nfs-ganesha-rados-
grace-2.8.3+git0.d504d374e-3.3.1.x86_64 ..........................................
[done]
(5/5) Removing nfs-
ganesha-2.8.3+git0.d504d374e-3.3.1.x86_64 ..........................................
[done]
Additional rpm output:
warning: /etc/ganesha/ganesha.conf saved as /etc/ganesha/
ganesha.conf.rpmsave
```

2. 从 Ceph Dashboard 中删除旧的集群设置：

```
cephuser@adm > ceph dashboard reset-ganesha-clusters-rados-pool-namespace
```

10.7.3 升级元数据服务器

与 MON、MGR 和 OSD 不同，无法就地采用元数据服务器。您需要使用 Ceph orchestrator 在容器中进行重新部署。

1. 运行 **`ceph fs ls`** 命令以获取您文件系统的名称，例如：

```
cephuser@adm > ceph fs ls
name: cephfs, metadata pool: cephfs_metadata, data pools: [cephfs_data ]
```

2. 通过使用文件系统名称作为 `service_id`，并指定将要运行 MDS 守护进程的主机，如第 8.3.3 节 “部署元数据服务器”中所述创建新服务规范文件 `mds.yml`。例如：

```
service_type: mds
service_id: cephfs
placement:
  hosts:
  - ses-min1
  - ses-min2
  - ses-min3
```

3. 运行 **`ceph orch apply -i mds.yml`** 命令以应用服务规范并启动 MDS 守护进程。

10.7.4 升级 iSCSI 网关

要升级 iSCSI 网关，您需要使用 Ceph orchestrator 在容器中重新部署该网关。如果您有多个 iSCSI 网关，则需要逐个重新部署，以减少服务停机时间。

1. 停止并禁用每个 iSCSI 网关节点上的现有 iSCSI 守护进程：

```
> sudo systemctl stop rbd-target-gw
> sudo systemctl disable rbd-target-gw
> sudo systemctl stop rbd-target-api
> sudo systemctl disable rbd-target-api
```

2. 按第 8.3.5 节 “部署 iSCSI 网关”中所述，为 iSCSI 网关创建服务规范。为此，您需要现有 `/etc/ceph/iscsi-gateway.cfg` 文件中的 `pool`、`trusted_ip_list` 和 `api_*` 设置。如果您启用了 SSL 支持 (`api_secure = true`)，则还需要 SSL 证书 (`/etc/ceph/iscsi-gateway.crt`) 和密钥 (`/etc/ceph/iscsi-gateway.key`)。
例如，如果 `/etc/ceph/iscsi-gateway.cfg` 包含以下配置：

```
[config]
cluster_client_name = client.igw.ses-min5
pool = iscsi-images
trusted_ip_list = 10.20.179.203,10.20.179.201,10.20.179.205,10.20.179.202
api_port = 5000
api_user = admin
api_password = admin
api_secure = true
```

则您需要创建以下服务规范文件 `iscsi.yml`：

```
service_type: iscsi
service_id: igw
placement:
  hosts:
  - ses-min5
spec:
  pool: iscsi-images
  trusted_ip_list:
 "10.20.179.203,10.20.179.201,10.20.179.205,10.20.179.202"
```

```
  api_port: 5000
  api_user: admin
  api_password: admin
  api_secure: true
  ssl_cert: |
    -----BEGIN CERTIFICATE-----
    MIIDtTCCAp2gAwIBAgIYMC4xNzc1NDQxNjEzMzc2MjMyXzxvQ7EcMA0GCSqGSIb3
    DQEBCwUAMG0xCzAJBgNVBAYTAlVTMQ0wCwYDVQQIDARVdGFoMRcwFQYDVQQHDA5T
    [...]
    -----END CERTIFICATE-----
  ssl_key: |
    -----BEGIN PRIVATE KEY-----
    MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQC5jdYbjtNTAKW4
    /CwQr/7wOiLGzVxChn3mmCIF3DwbL/qvTFTX2d8bDf6LjGwLYloXHscRfxszX/4h
    [...]
    -----END PRIVATE KEY-----
```

注意

`pool`、`trusted_ip_list`、`api_port`、`api_user`、`api_password`、`api_secure` 设置与 `/etc/ceph/iscsi-gateway.cfg` 文件中的相应设置相同。可从现有 SSL 证书和密钥文件中复制 `ssl_cert` 和 `ssl_key` 值。确认这些值是否缩进正确，并且 `ssl_cert:` 和 `ssl_key:` 行的末尾是否显示有**竖线**字符 `|`（请参见上述 `iscsi.yml` 文件的内容）。

3. 运行 **`ceph orch apply -i iscsi.yml`** 命令以应用服务规范并启动 iSCSI 网关守护进程。

4. 从每个现有 iSCSI 网关节点中删除旧的 `ceph-iscsi` 软件包：

```
cephuser@adm > zypper rm -u ceph-iscsi
```

10.8 升级后清理

升级后，请执行以下清理步骤：

1. 通过检查当前的 Ceph 版本，确认集群是否已成功升级：

```
cephuser@adm > ceph versions
```

2. 确保没有旧的 OSD 加入集群：

```
cephuser@adm > ceph osd require-osd-release pacific
```

3. 根据需要设置现有存储池的 `pg_autoscale_mode`：

重要

默认情况下，在 SUSE Enterprise Storage 6 中存储池的 `pg_autoscale_mode` 设置为 `warn`。这会导致在 PG 数量未达到最佳时出现警报消息，但实际上并未发生自动扩展。在 SUSE Enterprise Storage 7.1 中，新存储池的 `pg_autoscale_mode` 选项默认设为 `on`，因此 PG 实际上将自动扩展。而升级过程并不会自动更改现有存储池的 `pg_autoscale_mode`。如果您要将其更改为 `on` 以充分利用自动扩展器的功能，请参见《管理和操作指南》，第 17 章 “存储的数据管理”，第 17.4.12 节 “启用 PG 自动扩展器”中的说明。

有关详细信息，请参见《管理和操作指南》，第 17 章 “存储的数据管理”，第 17.4.12 节 “启用 PG 自动扩展器”。

4. 阻止早于 Luminous 版本的客户端：

```
cephuser@adm > ceph osd set-require-min-compat-client luminous
```

5. 启用平衡器模块：

```
cephuser@adm > ceph balancer mode upmap
cephuser@adm > ceph balancer on
```

有关详细信息，请参见《管理和操作指南》，第 29 章 “Ceph Manager 模块”，第 29.1 节 “平衡器”。

6. （可选）启用遥测模块：

```
cephuser@adm > ceph mgr module enable telemetry
```

```
cephuser@adm > ceph telemetry on
```

有关详细信息，请参见《管理和操作指南》, 第 29 章 “Ceph Manager 模块”, 第 29.2 节 “启用遥测模块”。

11 从 SUSE Enterprise Storage 7 升级到版本 7.1

本章说明将 SUSE Enterprise Storage 7 升级到版本 7.1 的步骤。

升级包括以下任务：

- 将底层 SUSE Linux Enterprise Server 15 SP2 升级到 SUSE Linux Enterprise Server 15 SP3 版本。
- 从 Ceph Octopus 升级到 Pacific。

11.1 升级前

开始升级之前，**必须**完成以下任务。在 SUSE Enterprise Storage 7 生命周期内可随时执行这些任务。

11.1.1 需考虑的要点

升级前，请务必通读以下内容，确保您了解所有需要执行的任务。

- **阅读发行说明**。在发行说明中，可以找到更多有关自 SUSE Enterprise Storage 的上一个版本发行后所进行的更改的信息。检查发行说明以了解：
  - 您的硬件是否有特殊注意事项。
  - 所使用的任何软件包是否发生了重大更改。
  - 是否需要对您的安装实施特殊预防措施。

  发行说明还提供未能及时编入手册中的信息。它们还包含有关已知问题的说明。
  您可以在 https://www.suse.com/releasenotes/ 上找到联机 SES 7.1 发行说明。
  此外，安装 SES 7.1 软件源中的 `release-notes-ses` 软件包之后，可在本地的 `/usr/share/doc/release-notes` 目录中或 https://www.suse.com/releasenotes/ 网页上找到发行说明。
- 阅读第 II 部分 “部署 Ceph 集群”，熟悉 `ceph-salt` 和 Ceph orchestrator，尤其要了解有关服务规范的信息。

11.1.2 备份集群配置和数据

我们强烈建议您在开始升级之前备份所有集群配置和数据。有关如何备份所有数据的说明，请参见《管理和操作指南》, 第 15 章 “备份和恢复”。

11.1.3 确认对软件源和容器映像的访问权限

确认是否每个集群节点都能访问 SUSE Linux Enterprise Server 15 SP3 和 SUSE Enterprise Storage 7.1 软件源以及容器映像注册表。

11.1.3.1 软件源

如果所有节点都已在 SCC 中注册，您便可以使用 `zypper migration` 命令进行升级。有关更多详细信息，请参见https://documentation.suse.com/sles/15-SP3/html/SLES-all/cha-upgrade-online.html#sec-upgrade-online-zypper。

如果节点**未**在 SCC 中注册，请禁用所有现有软件源，并为以下每个扩展添加 `Pool` 和 `Updates` 软件源：

- SLE-Product-SLES/15-SP3
- SLE-Module-Basesystem/15-SP3
- SLE-Module-Server-Applications/15-SP3
- SUSE-Enterprise-Storage-7.1

11.1.3.2 容器映像

所有集群节点都需要访问容器映像注册表。在大多数情况下，您将使用 `registry.suse.com` 上的公共 SUSE 注册表。您需要以下映像：

- registry.suse.com/ses/7.1/ceph/ceph
- registry.suse.com/ses/7.1/ceph/grafana
- registry.suse.com/ses/7.1/ceph/prometheus-server

- registry.suse.com/ses/7.1/ceph/prometheus-node-exporter
- registry.suse.com/ses/7.1/ceph/prometheus-alertmanager

或者，例如对于实体隔离部署，请配置本地注册表并确认您是否有一组正确的容器映像可用。有关配置本地容器映像注册表的更多详细信息，请参见第 7.2.10 节 “使用容器注册表”。

11.2 将每个集群节点上的 SUSE Linux Enterprise Server 迁移到 SUSE Linux Enterprise Server 15 SP3 版本

如果集群节点配置为使用 SUSE Customer Center，您可以使用 **`zypper migration`** 命令。

如果集群节点的软件源是手动配置的，则需要手动升级节点。

有关使用 **`zypper`** 升级 SUSE Linux Enterprise Server 的详细信息，请参见 https://documentation.suse.com/sles/15-SP3/html/SLES-all/cha-upgrade-online.html#sec-upgrade-online-zypper。

11.3 升级每个集群节点上与 SUSE Enterprise Storage 相关的软件包

要将 SUSE Enterprise Storage 软件包升级到最新版本，请使用 **`ceph-salt update`** 命令。有关详细信息，请参考 《管理和操作指南》, 第 13 章 “操作任务”, 第 13.6 节 “更新集群节点”。

11.4 升级现有 Ceph 集群服务

从管理节点中运行以下命令来将整个 Ceph 集群升级到 Pacific 版本：

```
cephuser@adm > ceph orch upgrade start --image registry.suse.com/ses/7.1/ceph/
ceph
```

A 基于上游“Pacific”小版本的 Ceph 维护更新

SUSE Enterprise Storage 7.1 中的几个关键软件包均基于 Ceph 的 Pacific 版本系列。当 Ceph 项目 (https://github.com/ceph/ceph) 在 Pacific 系列中发布新的小版本时，SUSE Enterprise Storage 7.1 即会更新，以确保产品应用最新的上游 Bug 修复并可进行功能向后移植。

本章简要介绍有关已经或计划在产品中包含的每个上游小版本中的重要更改。

词汇表

一般

Alertmanager

用于处理 Prometheus 服务器发送的告警并通知最终用户的单个二进制文件。

Ceph Dashboard

一个内置的基于 Web 的 Ceph 管理和监控应用，负责管理集群的各个方面和对象。仪表盘作为 Ceph Manager 模块实施。

Ceph Manager

Ceph Manager 或 MGR 是 Ceph Manager 软件，会将整个集群的所有状态收集到一处。

Ceph Monitor

Ceph Monitor 或 MON 是 Ceph Monitor 软件。

Ceph OSD 守护进程

`ceph-osd` 守护进程是 Ceph 的组件，负责在本地文件系统上存储对象并在网络中提供对它们的访问。

Ceph 存储集群

用于存储用户数据的存储软件的核心集合。此类集合由 Ceph Monitor 和 OSD 组成。

Ceph 客户端

可以访问 Ceph 存储集群的 Ceph 组件集合。其中包括对象网关、Ceph 块设备、CephFS 及其相应的库、内核模块和 FUSE 客户端。

Ceph 对象存储

对象存储“产品”、服务或功能，由 Ceph 存储集群和 Ceph 对象网关组成。

`ceph-salt`

提供使用 Salt 部署由 cephadm 管理的 Ceph 集群的工具。

cephadm

cephadm 用于部署和管理 Ceph 集群，它会通过 SSH 从 Manager 守护进程连接到主机以添加、删除或更新 Ceph 守护进程容器。

CephFS

Ceph 文件系统。

CephX

Ceph 身份验证协议。Cephx 操作与 Kerberos 类似，但它没有单一故障点。

CRUSH 规则

适用于某个特定存储池或多个存储池的 CRUSH 数据归置规则。

CRUSH、CRUSH 索引

基于可缩放哈希的受控复制：通过计算数据存储位置来确定如何存储和检索数据的算法。CRUSH 需要获取集群的索引来以伪随机的方式在 OSD 中存储和检索数据，并以一致的方式在整个集群中分布数据。

DriveGroups

DriveGroups 是可映射到物理驱动器的一个或多个 OSD 布局的声明。OSD 布局定义了 Ceph 如何在符合指定准则的媒体上实际分配 OSD 存储。

Grafana

数据库分析和监视解决方案。

OSD

对象存储设备：一个物理或逻辑存储单元。

OSD 节点

一个集群节点，用于存储数据、处理数据复制、恢复、回填、重新平衡以及通过检查其他 Ceph OSD 守护进程为 Ceph Monitor 提供某些监视信息。

PG

归置组：**存储池**的细分，用于性能调节。

Prometheus

系统监视和警告工具箱。

RADOS 块设备 (RBD)

Ceph 的块存储组件。也称为 Ceph 块设备。

Samba

Windows 集成软件。

Samba 网关

Samba 网关加入到 Windows 域中的 Active Directory，以进行用户身份验证和授权。

元数据服务器

元数据服务器或 MDS 是 Ceph 元数据软件。

区域组

可靠自主分布式对象存储 (RADOS)

用于存储用户数据的存储软件的核心集合 (MON+OSD)。

多区域

存储桶

将其他节点聚合成物理位置层次结构的一个点。

存档同步扩展模块

允许创建对象网关区域以保留 S3 对象版本历史记录的扩展模块。

对象网关

Ceph 对象存储的 S3/Swift 网关组件。也称为 RADOS 网关 (RGW)。

小版本

任何只包含 bug 或安全修复的临时版本。

池

用于存储磁盘映像等对象的逻辑分区。

管理节点

在其上运行与 Ceph 相关的命令以管理集群主机的主机。

节点

Ceph 集群中的任何一台计算机或服务器。

规则组

用于确定存储池的数据归置的规则。

路由树

该术语指的是展示接收器可运行的不同路由的任何示意图。