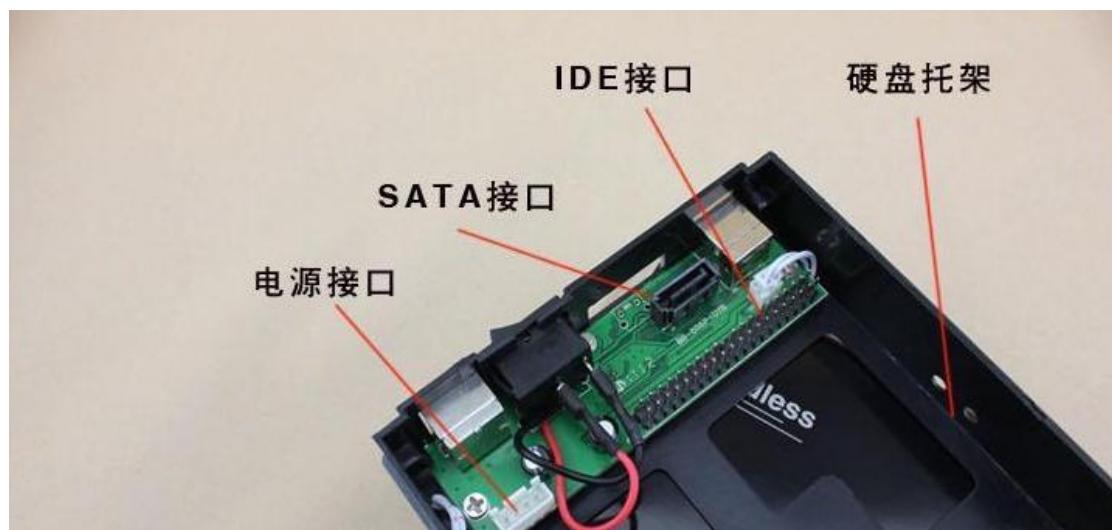


一：分布式存储概述：

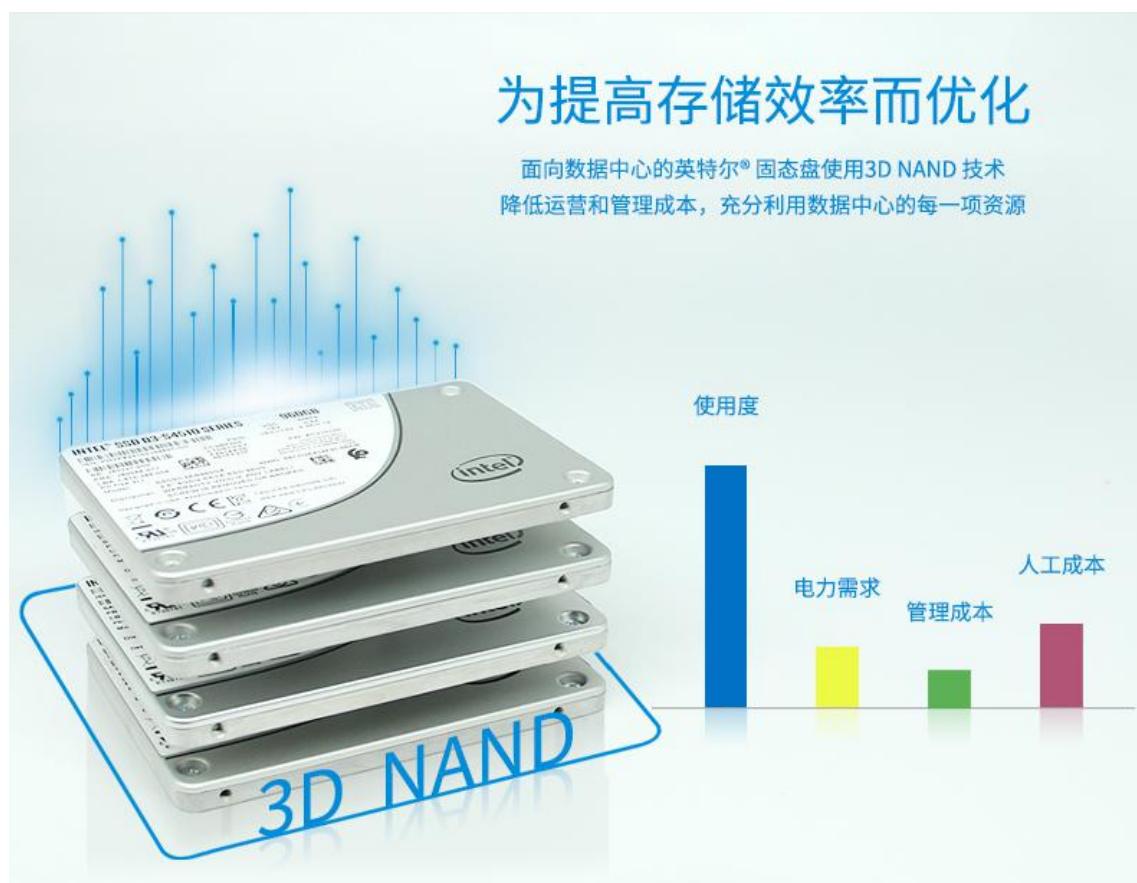
1.1:存储分类：

单机存储:

SCSI/IDE/SATA//SAS/USB/PCI-E/SSD/M.2 NVME 协议(提升性能)



<https://item.jd.com/49620677951.html#crumb-wrap>



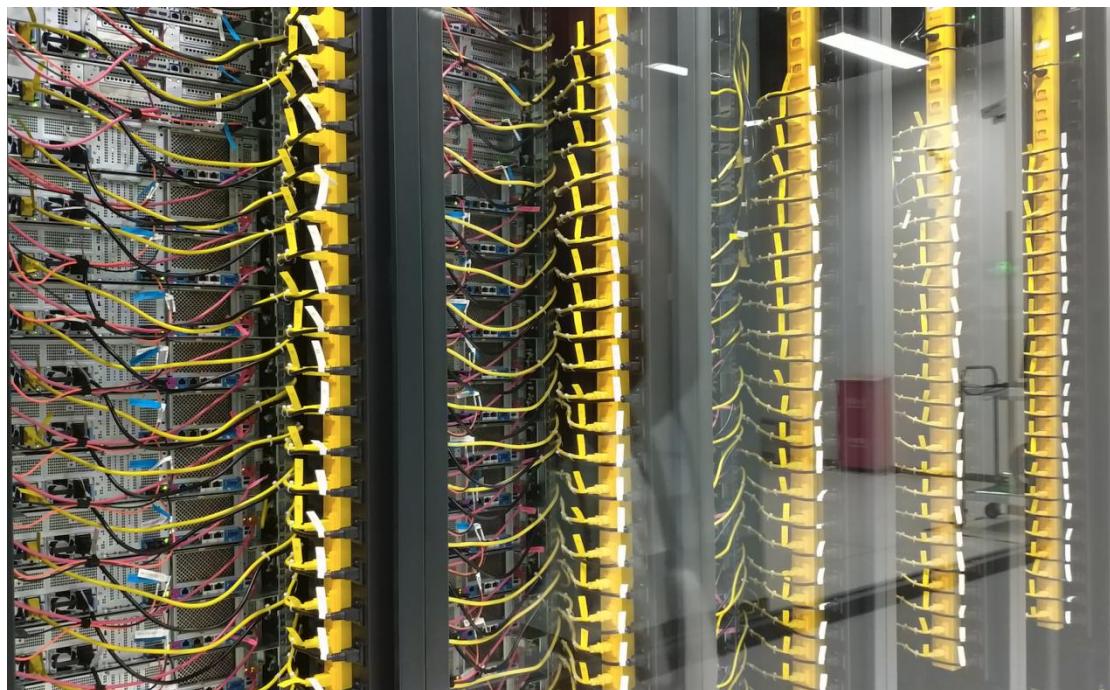
网络存储(带文件系统):

NFS

Samba

NAS (Network Attached Storage: 网络附属存储)

SAN:SAN (Storage Area Network, 存储区域网络)



存储考虑:

单机:

单机存储的磁盘空间问题、IO 问题、扩容问题、高可用问题:

商业:

商业解决方案-EMC、NetAPP、戴尔、华为、浪潮

分布式存储:

(软件定义的存储 Software Defined Storage SDS):

<https://www.vmware.com/cn/products/software-defined-storage.html>

分布式存储: Ceph, TFS, FastDFS, MogileFS, MooseFS, GlusterFS

对比说明 /文件系统	TFS	FastDFS	MogileF S	MooseFS	GlusterFS	Ceph
开发语言	C++	C	Perl	C	C	C++
开源协议	GPL V2	GPL V3	GPL	GPL V3	GPL V3	LGPL
数据存储方式	块	文件/Trunk	文件	块	文件/块	对象/文件/块
集群节点通信协 议	私有协议 (TCP)	私有协议 (TC P)	HTTP	私有协议 (TC P)	私有协议 (TCP) / RDAM (远程直接访问内存)	私有协议 (TC P)
专用元数据存储 点	占用NS	无	占用DB	占用MFS	无	占用MDS
在线扩容	支持	支持	支持	支持	支持	支持
冗余备份	支持	支持	-	支持	支持	支持

有状态集群数据读写特性:

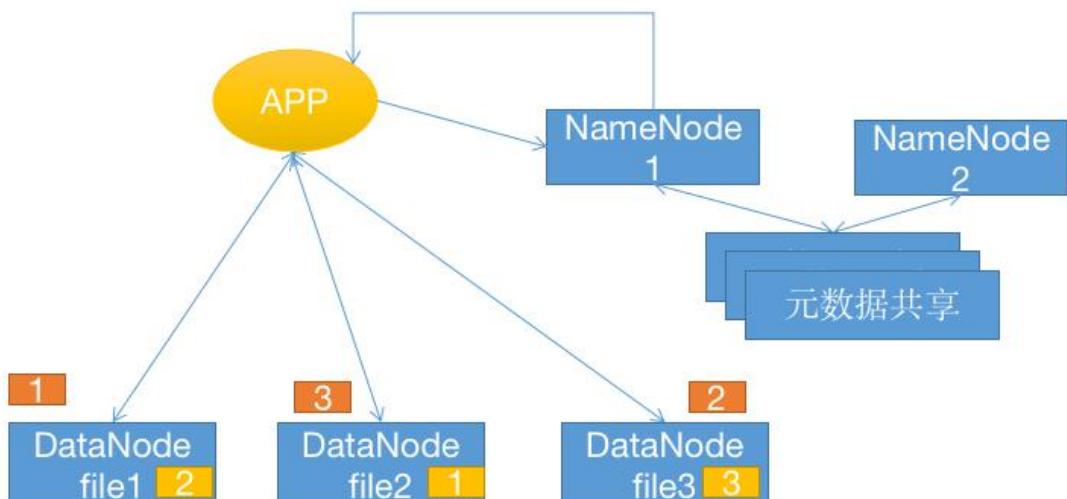
数据分为读数据和写数据, 读可以在任何一个节点读, 但是写只能写在特定的节点, 如 Redis 的 master、zookeeper 的 leader、MySQL 的 master 等场景。

1.2:分布式存储数据特性:

数据分为数据和元数据:

元数据即是文件的属性信息(文件名、权限(属主、属组)、大小、时间戳等), 在分布式存储中当客户端或者应用程序产生的客户端数据被写入到分布式存储系统的时候, 会有一个服务(Name Node)提供文件元数据的路由功能, 即告诉应用程序去哪个服务器去请求文件内容, 然后再有(Data Node)提供数据的读写请求及数据的高可用功能。

分布式存储



<https://www.redhat.com/zh/topics/data-storage/file-block-object-storage>

块存储: 需要格式化, 将文件直接保存到磁盘上。

文件存储: 提供数据存储的接口, 是由操作系统针对块存储的应用, 即由操作系统提供存储接口, 应用程序通过调用操作系统将文件保存到块存储进行持久化。

对象存储: 也称为基于对象的存储, 其中的文件被拆分成多个部分并散布在多个存储服务器, 在对象存储中, 数据会被分解为称为“对象”的离散单元, 并保存在单个存储库中, 而不是作为文件夹中的文件或服务器上的块来保存, 对象存储需要一个简单的 HTTP 应用编程接口 (API), 以供大多数客户端 (各种语言) 使用。

<https://www.aliyun.com/product/oss>

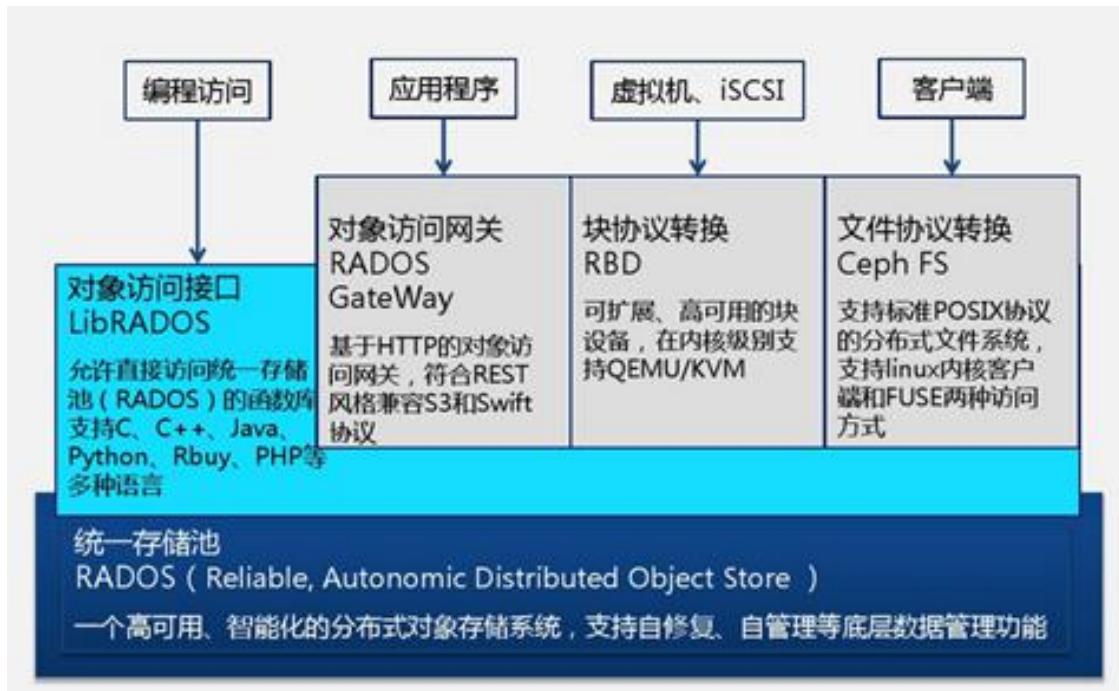
二:Ceph 基础:

Ceph 是一个开源的分布式存储系统, 同时支持对象存储、块设备、文件系统。

ceph 是一个对象(object)式存储系统, 它把每一个待管理的数据流(文件等数据)切分为一到多个固定大小(默认 4 兆)的对象数据, 并以其为原子单元(原子是构成元素的最小单元)完成数据的读写。

对象数据的底层存储服务是由多个存储主机(host)组成的存储集群, 该集群也被称为 RADOS(reliable automatic distributed object store)存储集群, 即可靠的、自动化的、分布式的对象存储系统。

librados 是 RADOS 存储集群的 API, 支持 C/C++/JAVA/python/ruby/php/go 等编程语言客户端。

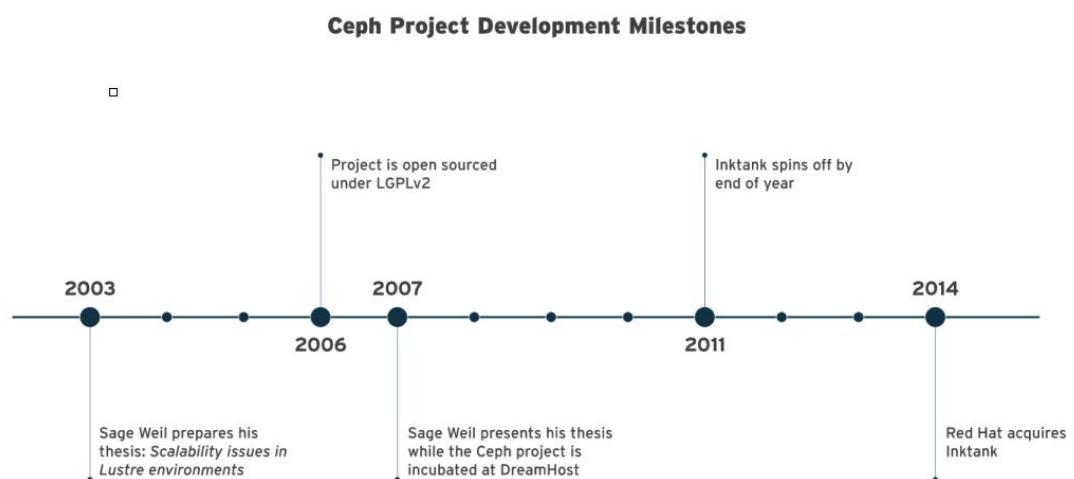


2.1:Ceph 的发展史:

Ceph 项目起源于于 2003 年在加州大学圣克鲁兹分校攻读博士期间的研究课题 (Lustre 环境中的可扩展问题) .

Lustre 是一种平行分布式文件系统,早在 1999 年,由皮特 · 布拉姆(Peter Braam)创建的集群文件系统公司(Cluster File Systems Inc)开始研发,并于 2003 年发布 Lustre 1.0 版本.

2007 年 Sage Weil(塞奇 · 威尔)毕业后, Sage Weil 继续全职从事 Ceph 工作, 2010 年 3 月 19 日, Linus Torvalds 将 Ceph 客户端合并到 2010 年 5 月 16 日发布的 Linux 内核版本 2.6.34,2012 年 Sage Weil 创建了 Inktank Storage 用于为 Ceph 提供专业服务和支持,2014 年 4 月 Redhat 以 1.75 亿美元收购 inktank 公司并开源.



2.2:ceph 的设计思想:

Ceph 的设计旨在实现以下目标:
每一组件皆可扩展
无单点故障
基于软件(而非专用设备)并且开源(无供应商锁定)
在现有的廉价硬件上运行
尽可能自动管理, 减少用户干预

2.3:ceph 的版本历史:

Ceph 的第一个版本是 0.1,发布日期为 2008 年 1 月,多年来 ceph 的版本号一直采用递归更新的方式没变,直到 2015 年 4 月 0.94.1(Hammer 的第一个修正版)发布后,为了避免 0.99(以及 0.100 或 1.00),后期的命名方式发生了改变:

x.0.z - 开发版 (给早期测试者和勇士们)

x.1.z - 候选版 (用于测试集群、高手们)

x.2.z - 稳定、修正版 (给用户们)

x 将从 9 算起, 它代表 Infernalis(首字母 I 是英文单词中的第九个字母),这样我们第九个发布周期的第一个开发版就是 9.0.0,后续的开发版依次是 9.0.0->9.0.1->9.0.2 等,测试版本就是 9.1.0->9.1.1->9.1.2,稳定版本就是 9.2.0->9.2.1->9.2.2.

[ceph / ceph](#)

The screenshot shows the GitHub interface for the 'ceph / ceph' repository. At the top, there are navigation links: Code, Pull requests (608), Actions, Projects (2), Security, and Insights. Below these are buttons for Go to file and Code. The main area has tabs for master, 50 branches, and 329 tags. A 'Switch branches/tags' dropdown menu is open, showing a list of branches and tags. The tags listed are v10.0.0, v9.2.1, v9.2.0, v9.1.0, v9.0.3, v9.0.2, v9.0.1, v9.0.0, v0.94.10, and v0.94.9. The first four (v9.2.1, v9.2.0, v9.1.0, v9.0.3) are highlighted with a red box. To the right of the dropdown, a list of recent commits is displayed, each with a commit ID, author, date, and a brief description.

到 2017 年底, Ceph 项目都采取每年发布两个稳定版本的做法,从 Jewel 版到 Nautilus 之前,Ceph 经历过一段时间的每间隔 9 个月发布一个新版本,Nautilus 版本开始改为每年春季 3 月份发布一个稳定版本,并提供长达 26 个月左右的后期版本更新.

The screenshot shows the Ceph Releases (index) page from the official documentation website. The page has a header with the Ceph logo and navigation links. On the left, there's a sidebar with a search bar and a list of topics under 'Ceph Releases (index)'.

ACTIVE RELEASES

The following Ceph releases are actively maintained and receive periodic backports and security fixes.

Name	Initial release	Latest	End of life (estimated)
Pacific	2021-03-31	16.2.5	2023-06-01
Octopus	2020-03-23	15.2.14	2022-06-01
Nautilus	2019-03-19	14.2.22	2021-06-01

ARCHIVED RELEASES

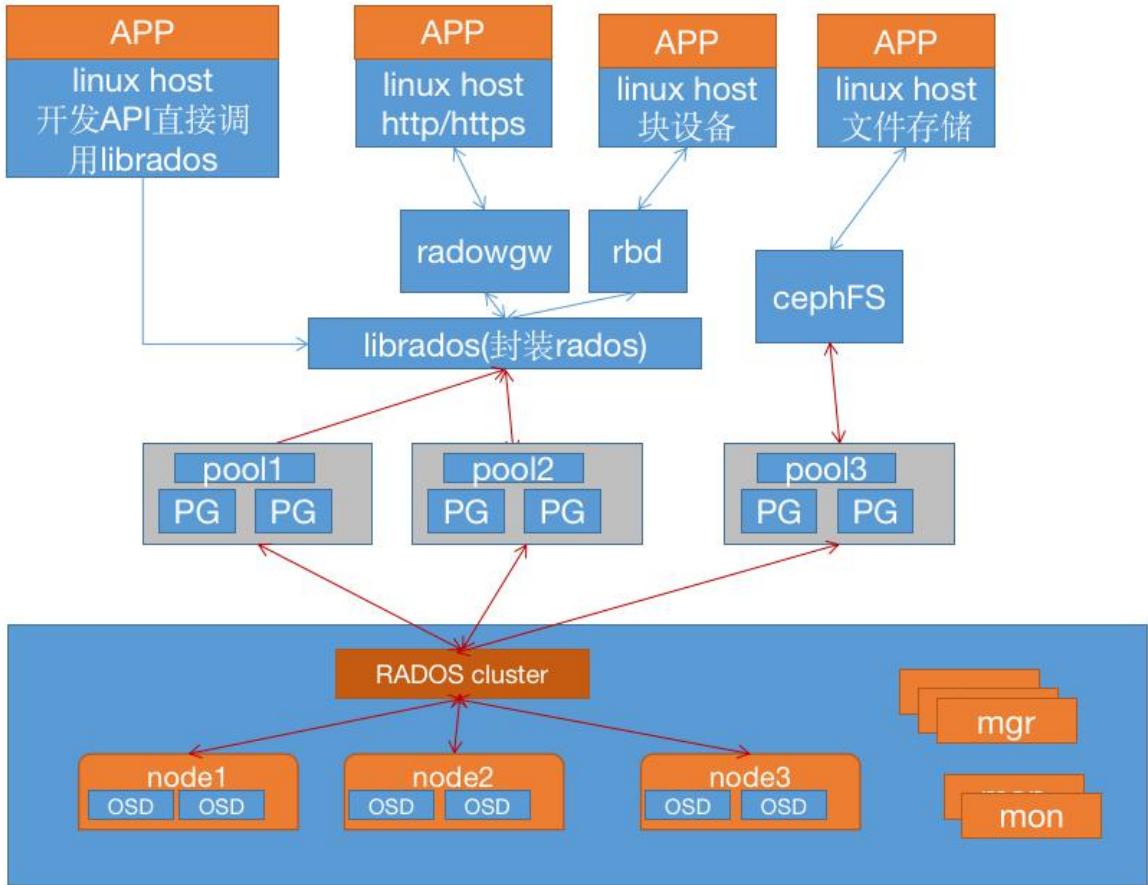
The following older Ceph releases are no longer maintained (do not receive bug fixes or backports).

Name	Initial release	Latest	End of life
Mimic	2018-06-01	13.2.10	2020-07-22
Luminous	2017-08-01	12.2.13	2020-03-01
Kraken	2017-01-01	11.2.1	2017-08-01
Jewel	2016-04-01	10.2.11	2018-07-01
Infernalis	2015-11-01	9.2.1	2016-04-01
Hammer	2015-04-01	0.94.10	2017-08-01
Giant	2014-10-01	0.87.2	2015-04-01
Firefly	2014-05-01	0.80.11	2016-04-01
Emperor	2013-11-01	0.72.2	2014-05-01
Dumpling	2013-08-01	0.67.11	2015-05-01

2.4:ceph 集群角色定义：

<https://docs.ceph.com/en/latest/start/intro/>

<http://docs.ceph.org.cn/start/intro/>



一个 ceph 集群的组成部分:

若干的 Ceph OSD (对象存储守护程序)

至少需要一个 Ceph Monitors 监视器 (1,3,5,7...)

两个或以上的 Ceph 管理器 managers, 运行 Ceph 文件系统客户端时

还需要高可用的 Ceph Metadata Server(文件系统元数据服务器)。

RADOS cluster:由多台 host 存储服务器组成的 ceph 集群

OSD(Object Storage Daemon): 每台存储服务器的磁盘组成的存储空间

Mon(Monitor): ceph 的监视器,维护 OSD 和 PG 的集群状态, 一个 ceph 集群至少要有一个 mon, 可以是一三五七等等这样的奇数个。

Mgr(Manager): 负责跟踪运行时指标和 Ceph 集群的当前状态, 包括存储利用率, 当前性能指标和系统负载等。

2.4.1:Monitor(ceph-mon) ceph 监视器:

在一个主机上运行的一个守护进程, 用于维护集群状态映射(maintains maps of the cluster state), 比如 ceph 集群中有多少存储池、每个存储池有多少 PG 以及存储池和 PG 的映射关系等, monitor map, manager map, the OSD map, the MDS map, and the CRUSH map, 这些映射是 Ceph 守护程序相互协调所需的关键群集状态, 此外监视器还负责管理守护程序和客户端之间的身份验证(认证使用 cephX 协议)。通常至少需要三个监视器才能实现冗余和高可用性。

2.4.2:Managers(ceph-mgr)的功能:

在一个主机上运行的一个守护进程，Ceph Manager 守护程序（ceph-mgr）负责跟踪运行时指标和 Ceph 集群的当前状态，包括存储利用率，当前性能指标和系统负载。Ceph Manager 守护程序还托管基于 python 的模块来管理和公开 Ceph 集群信息，包括基于 Web 的 Ceph 仪表板和 REST API。高可用性通常至少需要两个管理器。

2.4.3:Ceph OSDs(对象存储守护程序 ceph-osd):

提供存储数据，操作系统上的一个磁盘就是一个 OSD 守护程序，OSD 用于处理 ceph 集群数据复制，恢复，重新平衡，并通过检查其他 Ceph OSD 守护程序的心跳来向 Ceph 监视器和管理器提供一些监视信息。通常至少需要 3 个 Ceph OSD 才能实现冗余和高可用性。

2.4.4:MDS(ceph 元数据服务器 ceph-mds):

代表 ceph 文件系统(NFS/CIFS)存储元数据，(即 Ceph 块设备和 Ceph 对象存储不使用 MDS)

2.4.5:Ceph 的管理节点:

- 1.ceph 的常用管理接口是一组命令行工具程序，例如 rados、ceph、rbd 等命令，ceph 管理员可以从某个特定的 ceph-mon 节点执行管理操作
- 2.推荐使用部署专用的管理节点对 ceph 进行配置管理、升级与后期维护，方便后期权限管理，管理节点的权限只对管理人员开放，可以避免一些不必要的误操作的发生。

2.4.6:ceph 术语:

<http://docs.ceph.org.cn/glossary/>

2.5:ceph 逻辑组织架构:

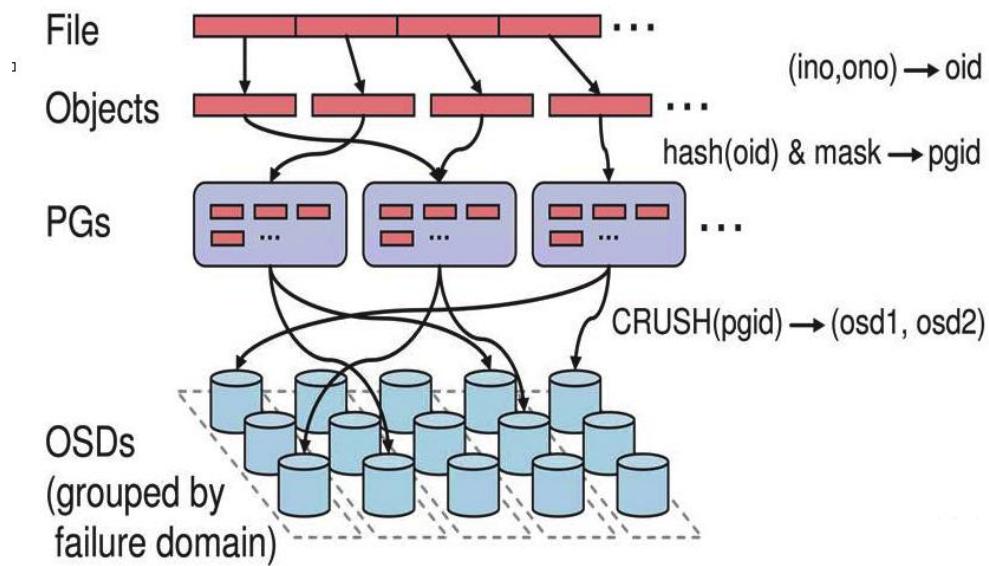
Pool: 存储池、分区，存储池的大小取决于底层的存储空间。

PG(placement group): 一个 pool 内部可以有多个 PG 存在，pool 和 PG 都是抽象的逻辑概念，一个 pool 中有多少个 PG 可以通过公式计算。

OSD(Object Storage Daemon, 对象存储设备): 每一块磁盘都是一个 osd，一个主机由一个或多个 osd 组成。

ceph 集群部署好之后，要先创建存储池才能向 ceph 写入数据，文件在向 ceph 保存之前要先进行一致性 hash 计算，计算后会把文件保存在某个对应的 PG 的，此文件一定属于某个 pool 的一个 PG，在通过 PG 保存在 OSD 上。

数据对象在写到主 OSD 之后再同步对从 OSD 以实现数据的高可用。



注：存储文件过程：

第一步：计算文件到对象的映射：

计算文件到对象的映射，假如 file 为客户端要读写的文件，得到 $\text{oid}(\text{object id}) = \text{ino} + \text{ono}$
 $\text{ino:inode number (INO)}$, File 的元数据序列号, File 的唯一 id。

$\text{ono:object number (ONO)}$, File 切分产生的某个 object 的序号, 默认以 4M 切分一个块大小。

第二步：通过 hash 算法计算出文件对应的 pool 中的 PG：

通过一致性 HASH 计算 Object 到 PG, Object -> PG 映射 $\text{hash}(\text{oid}) \& \text{mask} \rightarrow \text{pgid}$

第三步：通过 CRUSH 把对象映射到 PG 中的 OSD

通过 CRUSH 算法计算 PG 到 OSD, PG -> OSD 映射: $[\text{CRUSH}(\text{pgid}) \rightarrow (\text{osd1}, \text{osd2}, \text{osd3})]$

第四步：PG 中的主 OSD 将对象写入到硬盘

第五步：主 OSD 将数据同步给备份 OSD，并等待备份 OSD 返回确认

第六步：主 OSD 将写入完成返回给客户端

2.6:ceph 元数据保存方式：

Ceph 对象数据的元数据信息放在哪里呢？对象数据的元数据以 key-value 的形式存在，在 RADOS 中有两种实现：xattrs 和 omap：

ceph 可选后端支持多种存储引擎，比如 filestore, bluestore, kvstore, memstore, ceph 使用 bluestore 存储对象数据的元数据信息。

2.6.1:xattrs(扩展属性):

是将元数据保存在对象对应文件的扩展属性中并保存到系统磁盘上，这要求支持对象存储的本地文件系统（一般是 XFS）支持扩展属性。

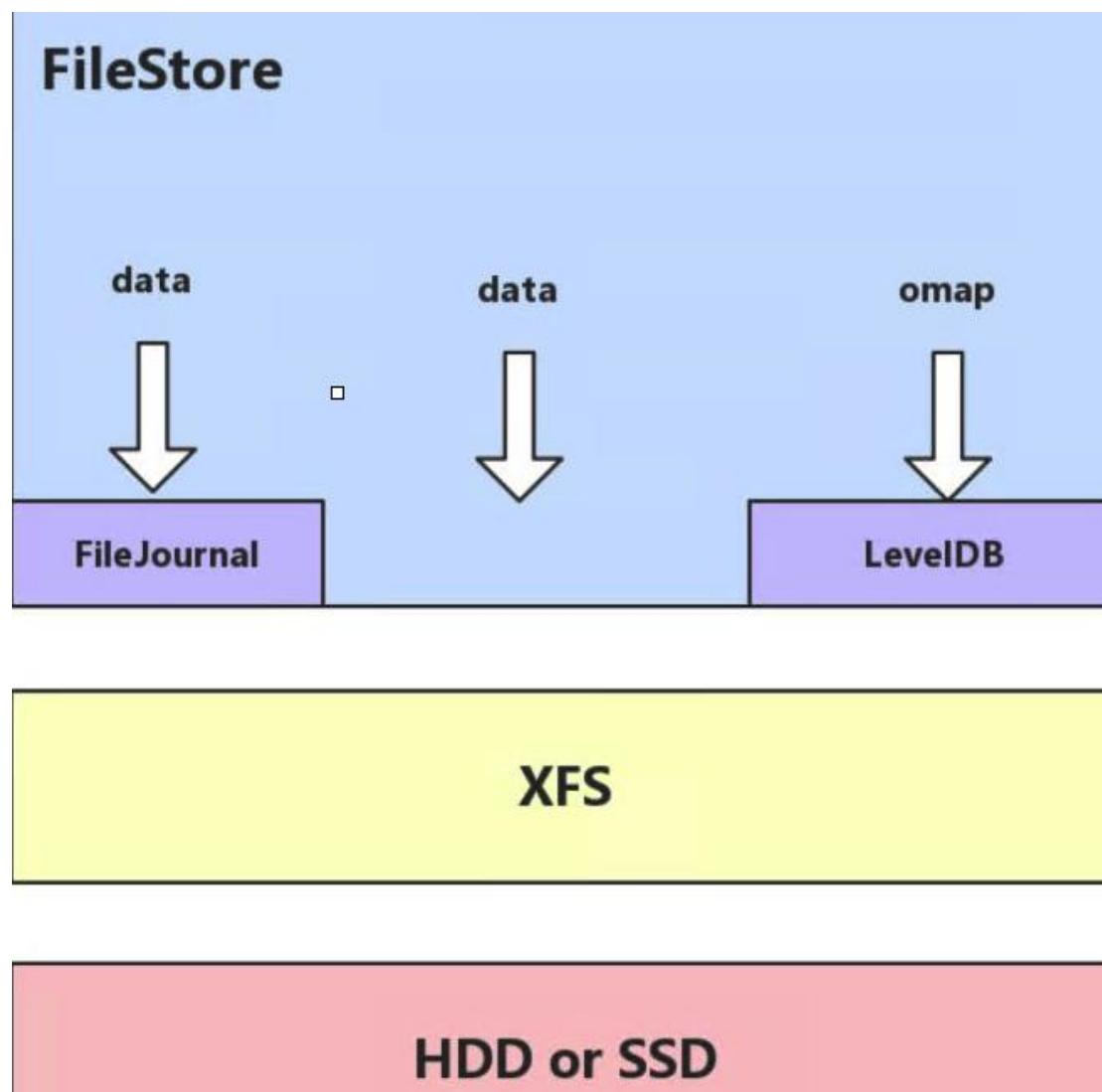
2.6.2:omap(object map 对象映射):

omap: 是 object map 的简称, 是将元数据保存在本地文件系统之外的独立 key-value 存储系统中, 在使用 filestore 时是 leveldb, 在使用 bluestore 时是 rocksdb, 由于 filestore 存在功能问题(需要将磁盘格式化为 XFS 格式)及元数据高可用问题等问题, 因此在目前 ceph 主要使用 bluestore。

2.6.2.1:filestore 与 leveldb:

ceph 早期基于 filestore 使用 google 的 leveldb 保存对象的元数据, LevelDb 是一个持久化存储的 KV 系统, 和 Redis 这种内存型的 KV 系统不同, leveldb 不会像 Redis 一样将数据放在内存从而占用大量的内存空间, 而是将大部分数据存储到磁盘上, 但是需要把磁盘上的 leveldb 空间格式化为文件系统(XFS).

FileStore 将数据保存到与 Posix 兼容的文件系统(例如 Btrfs、XFS、Ext4)。在 Ceph 后端使用传统的 Linux 文件系统尽管提供了一些好处, 但也有代价, 如性能、 对象属性与磁盘本地文件系统属性匹配存在限制等。



2.6.2.2:bluestore 与 rocksdb:

由于 levelDB 依然需要需要磁盘文件系统的支持，后期 facebook 对 levelDB 进行改进为 RocksDB <https://github.com/facebook/rocksdb>，RocksDB 将对象数据的元数据保存在 RocksDB，但是 RocksDB 的数据又放在哪里呢？放在内存怕丢失，放在本地磁盘但是解决不了高可用，ceph 对象数据放在了每个 OSD 中，那么就在在当前 OSD 中划分出一部分空间，格式化为 BlueFS 文件系统用于保存 RocksDB 中的元数据信息(称为 BlueStore)，并实现元数据的高可用，BlueStore 最大的特点是构建在裸磁盘设备之上，并且对诸如 SSD 等新的存储设备做了很多优化工作。

对全 SSD 及全 NVMe SSD 闪存适配

绕过本地文件系统层，直接管理裸设备，缩短 IO 路径

严格分离元数据和数据，提高索引效率

使用 KV 索引，解决文件系统目录结构遍历效率低的问题

支持多种设备类型

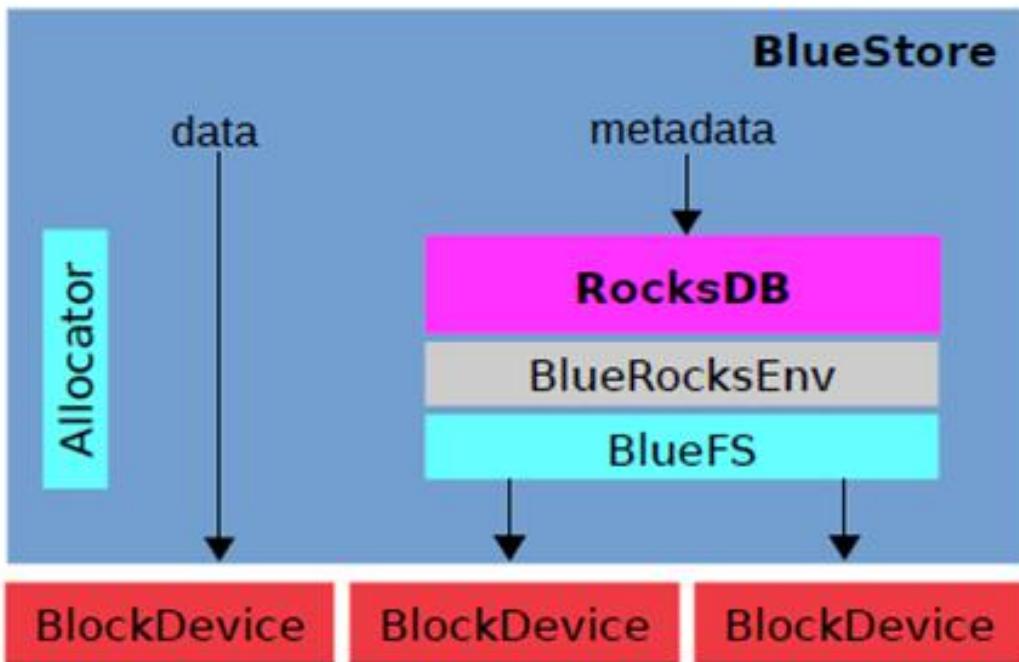
解决日志“双写”问题

期望带来至少 2 倍的写性能提升和同等读性能

增加数据校验及数据压缩等功能

RocksDB 通过中间层 BlueRocksDB 访问文件系统的接口。这个文件系统与传统的 Linux 文件系统（例如 Ext4 和 XFS）是不同的，它不是在 VFS 下面的通用文件系统，而是一个用户态的逻辑。BlueFS 通过函数接口（API，非 POSIX）的方式为 BlueRocksDB 提供类似文件系统的能力

RocksDB 通过中间层 BlueRocksDB 访问文件系统的接口。这个文件系统与传统的 Linux 文件系统（例如 Ext4 和 XFS）是不同的，它不是在 VFS 下面的通用文件系统，而是一个用户态的逻辑。BlueFS 通过函数接口（API，非 POSIX）的方式为 BlueRocksDB 提供类似文件系统的能力



BlueStore 的逻辑架构如上图所示，模块的划分都还比较清晰，我们来看下各模块的作用：

Allocator: 负责裸设备的空间管理分配。

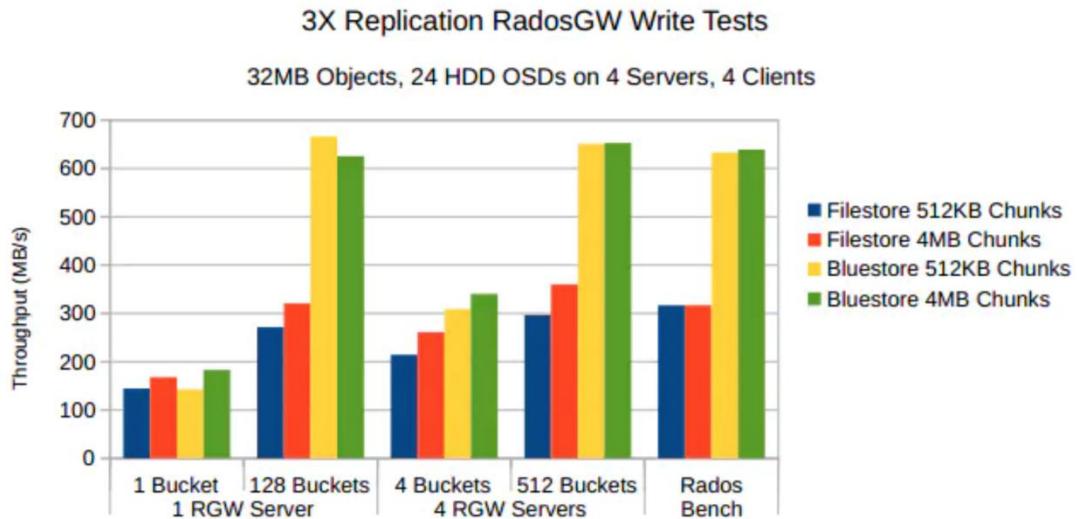
RocksDB: rocksdb 是 facebook 基于 leveldb 开发的一款 kv 数据库，BlueStore 将元数据全部存放至 RocksDB 中，这些元数据包括存储预写式日志、数据对象元数据、Ceph 的 omap 数据信息、以及分配器的元数据。

BlueRocksEnv: 这是 RocksDB 与 BlueFS 交互的接口；RocksDB 提供了文件操作的接口 EnvWrapper(Env 封装器)，可以通过继承实现该接口来自定义底层的读写操作，BlueRocksEnv 就是继承自 EnvWrapper 实现对 BlueFS 的读写。

BlueFS: BlueFS 是 BlueStore 针对 RocksDB 开发的轻量级文件系统，用于存放 RocksDB 产生的.sst 和.log 等文件。

BlockDevice: BlueStore 抛弃了传统的 ext4、xfs 文件系统，使用直接管理裸盘的方式；BlueStore 支持同时使用多种不同类型的设备，在逻辑上 BlueStore 将存储空间划分为三层：慢速 (Slow) 空间、高速 (DB) 空间、超高速 (WAL) 空间，不同的空间可以指定使用不同的设备类型，当然也可使用同一块设备。

BlueStore 的设计考虑了 FileStore 中存在的一些硬伤，抛弃了传统的文件系统直接管理裸设备，缩短了 IO 路径，同时采用 ROW 的方式，避免了日志双写的问题，在写入性能上有极大的提高。



2.7:Ceph CRUSH 算法简介：

Controllers replication under scalable hashing #可控的、可复制的、可伸缩的一致性 hash 算法。

Ceph 使用 CURSH 算法来存放和管理数据，它是 Ceph 的智能数据分发机制。Ceph 使用 CRUSH 算法来准确计算数据应该被保存到哪里，以及应该从哪里读取，和保存元数据不同的是，CRUSH 按需计算出元数据，因此它就消除了对中心式的服务器 / 网关的需求，它使得 Ceph 客户端能够计算出元数据，该过程也称为 CRUSH 查找，然后和 OSD 直接通信。

1.如果是把对象直接映射到 OSD 之上会导致对象与 OSD 的对应关系过于紧密和耦合，当 OSD 由于故障发生变更时将会对整个 ceph 集群产生影响。

2.于是 ceph 将一个对象映射到 RADOS 集群的时候分为两步走：

首先使用一致性 hash 算法将对象名称映射到 PG 2.7

然后将 PG ID 基于 CRUSH 算法映射到 OSD 即可查到对象

3.以上两个过程都是以“实时计算”的方式完成，而没有使用传统的查询数据与块设备的对应表的方式，这样有效避免了组件的“中心化”问题，也解决了查询性能和冗余问题。使得 ceph 集群扩展不再受查询的性能限制。

4.这个实时计算操作使用的就是 CRUSH 算法

Controllers replication under scalable hashing #可控的、可复制的、可伸缩的一致性 hash 算法。

CRUSH 是一种分布式算法，类似于一致性 hash 算法，用于为 RADOS 存储集群控制数据的分配。

三：部署 ceph 集群

<https://github.com/ceph/ceph>

[http://docs.ceph.org.cn/install/manual-deployment/ #简要部署过程](http://docs.ceph.org.cn/install/manual-deployment/)

版本历史：

<https://docs.ceph.com/en/latest/releases/index.html>

[https://docs.ceph.com/en/latest/releases/octopus/ #ceph 15 即 octopus 版本支持的系统：](https://docs.ceph.com/en/latest/releases/octopus/)

The screenshot shows the Ceph Octopus documentation page. The left sidebar has a dark background with white text. It includes links for Architecture, Developer Guide, Ceph Internals, Governance, Ceph Foundation, ceph-volume, and Ceph Releases (general). Under Ceph Releases (index), it lists Active Releases: Pacific (v16.2.*), Octopus (v15.2.*), v15.2.14 Octopus, v15.2.13 Octopus, v15.2.12 Octopus, v15.2.11 Octopus, v15.2.10 Octopus, v15.2.9 Octopus, and v15.2.8 Octopus. The main content area has a light blue header "V15.2.0 OCTOPUS". Below it, a teal box states "This is the first stable release of Ceph Octopus." A red box highlights the "MAJOR CHANGES FROM NAUTILUS" section, which contains a bulleted list of changes. Another red box highlights the "GENERAL" section, which contains a red watermark reading "马哥教育,杰哥的截图水印".

- A new deployment tool called `cephadm` has been introduced that integrates Ceph daemon deployment and management via containers into the orchestration layer. For more information see [Cephadm](#).
- Health alerts can now be muted, either temporarily or permanently.
- Health alerts are now raised for recent Ceph daemons crashes.
- A simple 'alerts' module has been introduced to send email health alerts for clusters deployed without the benefit of an existing external monitoring infrastructure.
- Packages are built for the following distributions:
 - CentOS 8
 - CentOS 7 (partial—see below)
 - Ubuntu 18.04 (Bionic)**
 - Debian Buster
 - Container image (based on CentOS 8)

[https://docs.ceph.com/en/latest/releases/pacific/ #ceph 16 即 Pacific 版本支持的系统：](https://docs.ceph.com/en/latest/releases/pacific/)

The screenshot shows the Ceph Pacific documentation page. The left sidebar has a dark background with white text. It includes links for Installing Ceph, Cephadm, Ceph Storage Cluster, Ceph File System, Ceph Block Device, Ceph Object Gateway, Ceph Manager Daemon, Ceph Dashboard, API Documentation, Architecture, Developer Guide, Ceph Internals, Governance, Ceph Foundation, ceph-volume, and Ceph Releases (general). The main content area has a light blue header "V16.2.0 PACIFIC". Below it, a teal box states "This is the first stable release of Ceph Pacific." A red box highlights the "MAJOR CHANGES FROM OCTOPUS" section, which contains a bulleted list of changes. Another red box highlights the "GENERAL" section, which contains a red watermark reading "马哥教育,杰哥的截图水印".

- Cephadm can automatically upgrade an Octopus cluster to Pacific with a single command to start the process.
- Cephadm has improved significantly over the past year, with improved support for RGW (standalone and multisite), and new support for NFS and iSCSI. Most of these changes have already been backported to recent Octopus point releases, but with the Pacific release we will switch to backporting bug fixes only.
- Packages are built for the following distributions:
 - CentOS 8
 - Ubuntu 20.04 (Focal)
 - Ubuntu 18.04 (Bionic)
 - Debian Buster
 - Container image (based on CentOS 8)

3.1:部署方式:

ceph-ansible: <https://github.com/ceph/ceph-ansible> #python

ceph-salt: <https://github.com/ceph/ceph-salt> #python

ceph-container: <https://github.com/ceph/ceph-container> #shell

ceph-chef: <https://github.com/ceph/ceph-chef> #Ruby

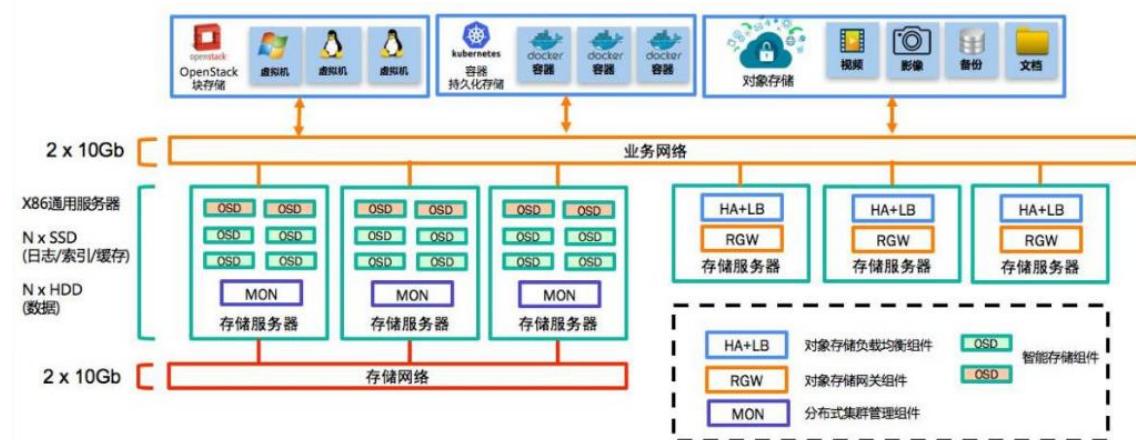
cephadm: <https://docs.ceph.com/en/latest/cephadm/> #ceph 官方在 ceph 15 版本加入的 ceph 部署工具

ceph-deploy: <https://github.com/ceph/ceph-deploy> #python

是一个 ceph 官方维护的基于 ceph-deploy 命令行部署 ceph 集群的工具，基于 ssh 执行可以 sudo 权限的 shell 命令以及一些 python 脚本 实现 ceph 集群的部署和管理维护。

Ceph-deploy 只用于部署和管理 ceph 集群，客户端需要访问 ceph，需要部署客户端工具。

3.2:服务器准备:



<http://docs.ceph.org.cn/start/hardware-recommendations/> #硬件推荐

1.四台服务器作为 ceph 集群 OSD 存储服务器，每台服务器支持两个网络，public 网络针对客户端访问，cluster 网络用于集群管理及数据同步，每台三块或以上的磁盘

172.31.6.106/192.168.6.106

172.31.6.107/192.168.6.107

172.31.6.108/192.168.6.108

172.31.6.109/192.168.6.109

各存储服务器磁盘划分:

/dev/sdb /dev/sdc /dev/sdd /dev/sde /dev/sdf #100G

2.三台服务器作为 ceph 集群 Mon 监视服务器，每台服务器可以和 ceph 集群的 cluster 网络通信。

172.31.6.101/192.168.6.101

172.31.6.102/192.168.6.102

172.31.6.103/192.168.6.103

3. 两个 ceph-mgr 管理服务器，可以和 ceph 集群的 cluster 网络通信。

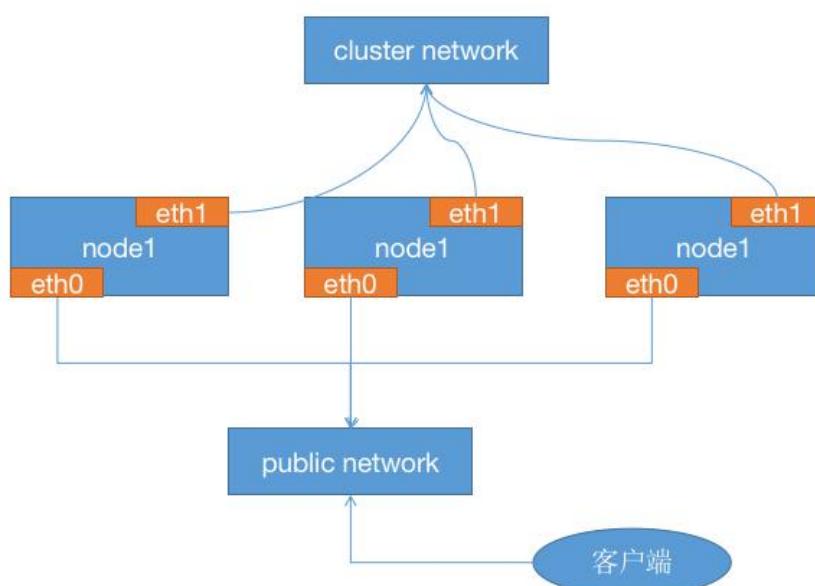
172.31.6.104/192.168.6.104

172.31.6.105/192.168.6.105

4. 一个服务器用于部署 ceph 集群即安装 Ceph-deploy，也可以和 ceph-mgr 等复用。

172.31.6.109/192.168.6.109

5. 创建一个普通用户，能够通过 sudo 执行特权命令，配置主机名解析，ceph 集群部署过程中需要对各主机配置不通的主机名，另外如果是 centos 系统则需要关闭各服务器的防火墙和 selinux。



3.3: 系统环境准备：

时间同步

关闭 selinux 和防火墙

配置域名解析或通过 DNS 解析

3.4: 部署 RADOS 集群：

<https://mirrors.aliyun.com/ceph/> #阿里云镜像仓库

<http://mirrors.163.com/ceph/> #网易镜像仓库

<https://mirrors.tuna.tsinghua.edu.cn/ceph/> #清华大学镜像源

3.4.1: 仓库准备：

各节点配置 ceph yum 仓库：

导入 key 文件：

```
~# wget -q -O- 'https://mirrors.tuna.tsinghua.edu.cn/ceph/keys/release.asc' | sudo apt-key add -
```

Centos 7.x:

```
[root@ceph-deploy ~]yum install https://mirrors.aliyun.com/ceph/rpm-octopus/el7/noarch/ceph-release-1-1.el7.noarch.rpm -y

[root@ceph-deploy ~]# ll /etc/yum.repos.d/
total 40
-rw-r--r--. 1 root root 1664 Apr  8 2020 CentOS-Base.repo
-rw-r--r--. 1 root root 1309 Apr  8 2020 CentOS-CR.repo
-rw-r--r--. 1 root root  649 Apr  8 2020 CentOS-Debuginfo.repo
-rw-r--r--. 1 root root  314 Apr  8 2020 CentOS-fasttrack.repo
-rw-r--r--. 1 root root  630 Apr  8 2020 CentOS-Media.repo
-rw-r--r--. 1 root root 1331 Apr  8 2020 CentOS-Sources.repo
-rw-r--r--. 1 root root 7577 Apr  8 2020 CentOS-Vault.repo
-rw-r--r--. 1 root root  616 Apr  8 2020 CentOS-x86_64-kernel.repo
-rw-r--r--  1 root root  535 May  5 2018 ceph.repo
```

各节点配置 epel 仓库:

```
[root@ceph-deploy ~]# yum install epel-release -y

[root@ceph-deploy ~]# ll /etc/yum.repos.d/
total 48
-rw-r--r--. 1 root root 1664 Apr  8 2020 CentOS-Base.repo
-rw-r--r--. 1 root root 1309 Apr  8 2020 CentOS-CR.repo
-rw-r--r--. 1 root root  649 Apr  8 2020 CentOS-Debuginfo.repo
-rw-r--r--. 1 root root  314 Apr  8 2020 CentOS-fasttrack.repo
-rw-r--r--. 1 root root  630 Apr  8 2020 CentOS-Media.repo
-rw-r--r--. 1 root root 1331 Apr  8 2020 CentOS-Sources.repo
-rw-r--r--. 1 root root 7577 Apr  8 2020 CentOS-Vault.repo
-rw-r--r--. 1 root root  616 Apr  8 2020 CentOS-x86_64-kernel.repo
-rw-r--r--  1 root root  535 May  5 2018 ceph.repo
-rw-r--r--  1 root root  951 Oct  3 2017 epel.repo
-rw-r--r--  1 root root 1050 Oct  3 2017 epel-testing.repo
```

如果仓库配置不正确，后期安装 ceph 会报错，如下：

```

----> Package python-six.noarch 0:1.9.0-2.el7 will be installed
--> Finished Dependency Resolution
Error: Package: 2:librgw2-13.2.10-0.el7.x86_64 (Ceph)
      Requires: liblttng-ust.so.0()(64bit)
Error: Package: 2:ceph-common-13.2.10-0.el7.x86_64 (Ceph)
      Requires: liblevelldb.so.1()(64bit)
Error: Package: 2:librgw2-13.2.10-0.el7.x86_64 (Ceph)
      Requires: liboauth.so.0()(64bit)
Error: Package: 2:ceph-common-13.2.10-0.el7.x86_64 (Ceph)
      Requires: liboath.so.0(LIBOATH_1.2.0)(64bit)
Error: Package: 2:ceph-common-13.2.10-0.el7.x86_64 (Ceph)
      Requires: liboath.so.0()(64bit)
Error: Package: 2:ceph-common-13.2.10-0.el7.x86_64 (Ceph)
      Requires: liboath.so.0(LIBOATH_1.10.0)(64bit)
Error: Package: 2:librbd1-13.2.10-0.el7.x86_64 (Ceph)
      Requires: liblttng-ust.so.0()(64bit)
Error: Package: 2:ceph-common-13.2.10-0.el7.x86_64 (Ceph)
      Requires: libbabeltrace-ctf.so.1()(64bit)
Error: Package: 2:ceph-common-13.2.10-0.el7.x86_64 (Ceph)
      Requires: libbabeltrace.so.1()(64bit)
Error: Package: 2:librados2-13.2.10-0.el7.x86_64 (Ceph)
      Requires: liblttng-ust.so.0()(64bit)
You could try using --skip-broken to work around the problem
You could try running: rpm -Va --nofiles --nodigest
[ceph@ceph-deploy ~]$ exit

```

Ubuntu 18.04.x:

```

~# cat /etc/apt/sources.list
# 默认注释了源码镜像以提高 apt update 速度，如有需要可自行取消注释
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-updates main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-updates main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-backports main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-backports main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-security main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-security main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ceph/debian-pacific bionic main

# sudo echo "deb https://mirrors.tuna.tsinghua.edu.cn/ceph/debian-pacific bionic main"
>> /etc/apt/sources.list

```

3.4.2:创建 ceph 用户:

推荐使用指定的普通用户部署和运行 **ceph** 集群，普通用户只要能以非交互方式执行 **sudo** 命令执行一些特权命令即可，新版的 **ceph-deploy** 可以指定包含 **root** 的在内只要可以执行 **sudo** 命令的用户，不过仍然推荐使用普通用户，比如 **ceph**、**cephuser**、**cephadmin** 这样的用户去管理 **ceph** 集群。

在包含 **ceph-deploy** 节点的存储节点、**mon** 节点和 **mgr** 节点等创建 **ceph** 用户。

```
[root@ceph-deploy ~]# groupadd -r -g 2022 ceph && useradd -r -m -s /bin/bash -u 2022 -g 2022 ceph && echo ceph:123456 | chpasswd
```

Centos:

```
[root@ceph-deploy ~]# groupadd ceph -g 2020 && useradd -u 2020 -g 2020 ceph && echo "123456" | passwd --stdin ceph
```

各服务器允许 ceph 用户以 sudo 执行特权命令:

```
# vim /etc/sudoers
root    ALL=(ALL)      ALL
ceph    ALL=(ALL)      NOPASSWD: ALL
```

或者:

```
~# echo "ceph    ALL=(ALL)      NOPASSWD: ALL" >> /etc/sudoers
```

配置免秘钥登录:

在 ceph-deploy 节点配置允许以非交互的方式登录到各 ceph node/mon/mgr 节点，即在 ceph-deploy 节点生成秘钥对，然后分发公钥到各被管理节点:

```
[root@ceph-deploy ~]# su - ceph
[ceph@ceph-deploy ~]$ ssh-keygen
 6 ssh-copy-id ceph@172.31.6.100
 7 ssh-copy-id ceph@172.31.6.101
 8 ssh-copy-id ceph@172.31.6.102
 9 ssh-copy-id ceph@172.31.6.103
10 ssh-copy-id ceph@172.31.6.104
11 ssh-copy-id ceph@172.31.6.105
12 ssh-copy-id ceph@172.31.6.106
13 ssh-copy-id ceph@172.31.6.107
14 ssh-copy-id ceph@172.31.6.108
15 ssh-copy-id ceph@172.31.6.109
```

3.4.3:配置主机名解析:

```
[ceph@ceph-deploy ~]$ vim /etc/hosts
172.31.6.100 ceph-deploy.example.local  ceph-deploy
172.31.6.101 ceph-mon1.example.local    ceph-mon1
172.31.6.102 ceph-mon2.example.local    ceph-mon2
172.31.6.103 ceph-mon3.example.local    ceph-mon3
172.31.6.104 ceph-mgr1.example.local   ceph-mgr1
172.31.6.105 ceph-mgr2.example.local   ceph-mgr2
172.31.6.106 ceph-node1.example.local  ceph-node1
172.31.6.107 ceph-node2.example.local  ceph-node2
172.31.6.108 ceph-node3.example.local  ceph-node3
172.31.6.109 ceph-node4.example.local  ceph-node4
```

3.4.4:安装 ceph 部署工具:

在 ceph 部署服务器安装部署工具 ceph-deploy

Ubuntu:

```
ceph@ceph-deploy:~$ apt-cache madison ceph-deploy
ceph-deploy | 2.0.1 | https://mirrors.tuna.tsinghua.edu.cn/ceph/debian-pacific bionic/main amd64
 Packages
ceph-deploy | 2.0.1 | https://mirrors.tuna.tsinghua.edu.cn/ceph/debian-pacific bionic/main i386
 Packages
ceph-deploy | 1.5.38-0ubuntu1 | https://mirrors.tuna.tsinghua.edu.cn/ubuntu bionic/universe amd64
 Packages
ceph-deploy | 1.5.38-0ubuntu1 | https://mirrors.tuna.tsinghua.edu.cn/ubuntu bionic/universe i386
 Packages
ceph@ceph-deploy:~$ sudo apt install ceph-deploy
```

Centos:

```
[ceph@ceph-deploy ~]$ sudo yum install ceph-deploy python-setuptools
python2-subprocess3
=====
Installing:
  ceph-deploy                               noarch          2.0.1-0
  python-setuptools                         noarch          0.9.8-7.el7
  python2-subprocess32                      x86_64         3.2.6-14.el7
Installing for dependencies:
  python-backports                          x86_64         1.0-8.el7
  python-backports-ssl_match_hostname        noarch          3.5.0.1-1.el7
  python-ipaddress                           noarch          1.0.16-2.el7
=====
Transaction Summary
=====
Install 3 Packages (+3 Dependent packages)

Total download size: 782 k
Installed size: 3.6 M
Is this ok [y/d/N]: y
```

3.4.5:初始化 mon 节点:

在管理节点初始化 mon 节点

```
[ceph@ceph-deploy ~]$ mkdir ceph-cluster #保存当前集群的初始化配置信息
[ceph@ceph-deploy ~]$ cd ceph-cluster/
```

<http://docs.ceph.org.cn/man/8/ceph-deploy/#id3>

```
$ ceph-deploy --help
new: 开始部署一个新的 ceph 存储集群，并生成 CLUSTER.conf 集群配置文件和 keyring 认证文件。
install: 在远程主机上安装 ceph 相关的软件包，可以通过--release 指定安装的版本。
rgw: 管理 RGW 守护程序(RADOSGW, 对象存储网关)。
mgr: 管理 MGR 守护程序(ceph-mgr,Ceph Manager DaemonCeph 管理器守护程序)。
mds: 管理 MDS 守护程序(Ceph Metadata Server, ceph 源数据服务器)。
mon: 管理 MON 守护程序(ceph-mon,ceph 监视器)。
gatherkeys: 从指定获取提供新节点的验证 keys, 这些 keys 会在添加新的 MON/OSD/MD
```

加入的时候使用。

disk: 管理远程主机磁盘。

osd: 在远程主机准备数据磁盘，即将指定远程主机的指定磁盘添加到 ceph 集群作为 osd 使用。

repo: 远程主机仓库管理。

admin: 推送 ceph 集群配置文件和 client.admin 认证文件到远程主机。

config: 将 ceph.conf 配置文件推送到远程主机或从远程主机拷贝。

uninstall: 从远端主机删除安装包。

purgedata: 从 /var/lib/ceph 删除 ceph 数据,会删除/etc/ceph 下的内容。

purge: 删除远端主机的安装包和所有数据。

forgetkeys: 从本地主机删除所有的验证 keyring, 包括 client.admin, monitor, bootstrap 等认证文件。

pkg: 管理远端主机的安装包。

calamari: 安装并配置一个 calamari web 节点, calamari 是一个 web 监控平台。

初始化 mon 节点过程如下：

Ubuntu 各服务器需要单独安装 Python2:

```
root@ceph-mon1:~# apt install python2.7 -y
root@ceph-mon1:~# ln -sv /usr/bin/python2.7 /usr/bin/python2
```

```
ceph@ceph-deploy:~/ceph-cluster$ ceph-deploy new --cluster-network 192.168.0.0/21 --public-network 172.31.0.0/21 ceph-mon2.example.local
[ceph_deploy.conf][DEBUG ] found configuration file at: /home/ceph/.cephdeploy.conf
[ceph_deploy.cli][INFO  ] Invoked (2.0.1): /usr/bin/ceph-deploy new --cluster-network 192.168.0.0/21 --public-network 172.31.0.0/21 ceph-mon2.example.local
[ceph_deploy.cli][INFO  ] ceph-deploy options:
[ceph_deploy.cli][INFO  ]   username          : None
[ceph_deploy.cli][INFO  ]   verbose           : False
[ceph_deploy.cli][INFO  ]   overwrite_conf    : False
[ceph_deploy.cli][INFO  ]   quiet              : False
[ceph_deploy.cli][INFO  ]   cd_conf             : <ceph_deploy.conf.cephdeploy.Conf instance at 0x7fdf77eb1dc0>
[ceph_deploy.cli][INFO  ]   cluster             : ceph
[ceph_deploy.cli][INFO  ]   ssh_copykey       : True
[ceph_deploy.cli][INFO  ]   mon                : ['ceph-mon2.example.local']
[ceph_deploy.cli][INFO  ]   func               : <function new at 0x7fdf75168ad0>
[ceph_deploy.cli][INFO  ]   public_network     : 172.31.0.0/21
[ceph_deploy.cli][INFO  ]   ceph_conf          : None
[ceph_deploy.cli][INFO  ]   cluster_network     : 192.168.0.0/21
[ceph_deploy.cli][INFO  ]   default_release    : False
[ceph_deploy.cli][INFO  ]   fsid               : None
[ceph_deploy.new][DEBUG ] Creating new cluster named ceph
[ceph_deploy.new][INFO  ] making sure passwordless SSH succeeds
[ceph-mon2.example.local][DEBUG ] connected to host: ceph-deploy.example.local
[ceph-mon2.example.local][INFO  ] Running command: ssh -CT -o BatchMode=yes ceph-mon2.example.local
bash: python2: command not found
[ceph_deploy][ERROR ] RuntimeError: connecting to host: ceph-mon2.example.local resulted in errors: IOError cannot send (already closed?)
```

马哥教育, 杰哥的截图水印

```
[ceph@ceph-deploy  ceph-cluster]$ ceph-deploy new --cluster-network
192.168.0.0/21 --public-network 172.31.0.0/21 ceph-mon1.example.local
[ceph_deploy.conf][DEBUG ] found configuration file at: /home/ceph/.cephdeploy.conf
[ceph_deploy.cli][INFO  ] Invoked (2.0.1): /usr/bin/ceph-deploy new --cluster-network 192.168.0.0/21 --public-network
172.31.0.0/21 ceph-mon1.example.local
[ceph_deploy.cli][INFO  ] ceph-deploy options:
[ceph_deploy.cli][INFO  ]   username          : None
[ceph_deploy.cli][INFO  ]   verbose           : False
[ceph_deploy.cli][INFO  ]   overwrite_conf    : False
[ceph_deploy.cli][INFO  ]   quiet              : False
[ceph_deploy.cli][INFO  ]   cd_conf             : <ceph_deploy.conf.cephdeploy.Conf instance at
0x7f1a23926dc0>
[ceph_deploy.cli][INFO  ]   cluster             : ceph
[ceph_deploy.cli][INFO  ]   ssh_copykey       : True
```

```
[ceph_deploy.cli][INFO ] mon : ['ceph-mon1.example.local']
[ceph_deploy.cli][INFO ] func : <function new at 0x7f1a20bddad0>
[ceph_deploy.cli][INFO ] public_network : 172.31.0.0/21
[ceph_deploy.cli][INFO ] ceph_conf : None
[ceph_deploy.cli][INFO ] cluster_network : 192.168.0.0/21
[ceph_deploy.cli][INFO ] default_release : False
[ceph_deploy.cli][INFO ] fsid : None
[ceph_deploy.new][DEBUG ] Creating new cluster named ceph
[ceph_deploy.new][INFO ] making sure passwordless SSH succeeds
[ceph-mon1.example.local][DEBUG ] connected to host: ceph-deploy.example.local
[ceph-mon1.example.local][INFO ] Running command: ssh -CT -o BatchMode=yes ceph-mon1.example.local
[ceph-mon1.example.local][DEBUG ] connection detected need for sudo
[ceph-mon1.example.local][DEBUG ] connected to host: ceph-mon1.example.local
[ceph-mon1.example.local][DEBUG ] detect platform information from remote host
[ceph-mon1.example.local][DEBUG ] detect machine type
[ceph-mon1.example.local][DEBUG ] find the location of an executable
[ceph-mon1.example.local][INFO ] Running command: sudo /bin/ip link show
[ceph-mon1.example.local][INFO ] Running command: sudo /bin/ip addr show
[ceph-mon1.example.local][DEBUG ] IP addresses found: [u'192.168.6.101', u'172.31.6.101']
[ceph_deploy.new][DEBUG ] Resolving host ceph-mon1.example.local
[ceph_deploy.new][DEBUG ] Monitor ceph-mon1 at 172.31.6.101
[ceph_deploy.new][DEBUG ] Monitor initial members are ['ceph-mon1']
[ceph_deploy.new][DEBUG ] Monitor addrs are [u'172.31.6.101']
[ceph_deploy.new][DEBUG ] Creating a random mon key...
[ceph_deploy.new][DEBUG ] Writing monitor keyring to ceph.mon.keyring...
[ceph_deploy.new][DEBUG ] Writing initial config to ceph.conf...
```

mon 节点也需要有 cluster network, 否则初始化报错:

```
[ceph-mon1.example.local][DEBUG ] IP addresses found: [u'172.31.6.101']
[ceph_deploy][ERROR ] RuntimeError: subnet (192.168.0.0/21) is not valid for any of the
ips found [u'172.31.6.101']
```

验证初始化:

```
[ceph@ceph-deploy ceph-cluster]$ ll
total 16
-rw-rw-r-- 1 ceph ceph 263 Oct 21 14:24 ceph.conf #自动生成的配置文件
-rw-rw-r-- 1 ceph ceph 7096 Oct 21 14:24 ceph-deploy-ceph.log #初始化日志
-rw----- 1 ceph ceph 73 Oct 21 14:24 ceph.mon.keyring #用于 ceph mon 节点内部通讯认证的秘钥环文件
```

```
[ceph@ceph-deploy ceph-cluster]$ cat ceph.conf
[global]
fsid = 80a34e06-4458-41a8-8d19-1c0501152d69 #ceph 的集群 ID
public_network = 172.31.0.0/21
cluster_network = 192.168.0.0/21
```

```
mon_initial_members = ceph-mon1 #可以用逗号做分割添加多个 mon 节点
mon_host = 172.31.6.104
auth_cluster_required = cephx
auth_service_required = cephx
auth_client_required = cephx
```

3.4.6:初始化 ceph 存储节点:

初始化存储节点等于在存储节点安装了 ceph 及 ceph-rodsgw 安装包，但是使用默认的官方仓库会因为网络原因导致初始化超时，因此各存储节点推荐修改 ceph 仓库为阿里或者清华等国内的镜像源：

3.4.6.1:修改 ceph 镜像源：

各节点配置清华的 ceph 镜像源：

3.4.6.1.1:ceph-mimic 版本(13):

```
# vim /etc/yum.repos.d/ceph.repo
[Ceph]
name=Ceph packages for $basearch
baseurl=https://mirrors.tuna.tsinghua.edu.cn/ceph/rpm-octopus/el7/$basearch
enabled=1
gpgcheck=1
type=rpm-md
gpgkey=https://mirrors.tuna.tsinghua.edu.cn/ceph/keys/release.asc

[Ceph-noarch]
name=Ceph noarch packages
baseurl=https://mirrors.tuna.tsinghua.edu.cn/ceph/rpm-octopus/el7/noarch
enabled=1
gpgcheck=1
type=rpm-md
gpgkey=https://mirrors.tuna.tsinghua.edu.cn/ceph/keys/release.asc

[ceph-source]
name=Ceph source packages
baseurl=https://mirrors.tuna.tsinghua.edu.cn/ceph/rpm-octopus/el7/SRPMS
enabled=1
gpgcheck=1
type=rpm-md
gpgkey=https://mirrors.tuna.tsinghua.edu.cn/ceph/keys/release.asc
```

3.4.6.1.2:ceph-nautilus 版本(14):

```
# vim /etc/yum.repos.d/ceph.repo
```

```
[Ceph]
name=Ceph packages for $basearch
```

```
baseurl=https://mirrors.tuna.tsinghua.edu.cn/ceph/rpm-nautilus/el7/$basearch
enabled=1
gpgcheck=1
type=rpm-md
gpgkey=https://mirrors.tuna.tsinghua.edu.cn/ceph/keys/release.asc

[Ceph-noarch]
name=Ceph noarch packages
baseurl=https://mirrors.tuna.tsinghua.edu.cn/ceph/rpm-nautilus/el7/noarch
enabled=1
gpgcheck=1
type=rpm-md
gpgkey=https://mirrors.tuna.tsinghua.edu.cn/ceph/keys/release.asc

[ceph-source]
name=Ceph source packages
baseurl=https://mirrors.tuna.tsinghua.edu.cn/ceph/rpm-nautilus/el7/SRPMS
enabled=1
gpgcheck=1
type=rpm-md
gpgkey=https://mirrors.tuna.tsinghua.edu.cn/ceph/keys/release.asc
```

3.4.6.1.3:ceph-octopus 版本(15):

Ubuntu 18.04:

```
~# cat /etc/apt/sources.list
# 默认注释了源码镜像以提高 apt update 速度，如有需要可自行取消注释
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-updates main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-updates main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-backports main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-backports main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-security main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-security main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ceph/debian-octopus bionic main
```

3.4.6.1.3:ceph-pacific 版本(16):

Ubuntu 18.04:

```
~# cat /etc/apt/sources.list
```

```
# 默认注释了源码镜像以提高 apt update 速度，如有需要可自行取消注释
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-updates main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-updates main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-backports main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-backports main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-security main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-security main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ceph/debian-pacific bionic main
```

3.4.6.2:修改 epel 镜像源：

<https://mirrors.tuna.tsinghua.edu.cn/help/epel/>

```
# yum install epel-release -y
# sed -e 's!^metalink=!=#metalink=!g' \
    -e 's!^#baseurl!=!baseurl=!g' \
    -e 's!//download\.fedoraproject\.org/pub//mirrors.tuna.tsinghua.edu.cn!g' \
    -e 's!http://mirrors\.tuna!https://mirrors.tuna!g' \
    -i /etc/yum.repos.d/epel.repo /etc/yum.repos.d/epel-testing.repo

# yum makecache fast
```

初始化 node 节点过程：

[ceph@ceph-deploy ceph-cluster]\$ ceph-deploy install --no-adjust-repos --nogpgcheck ceph-node1 ceph-node2 ceph-node3

```
[ceph-node4] [DEBUG ] Setting up radosgw (16.2.5-1bionic) ...
[ceph-node4] [DEBUG ] Created symlink /etc/systemd/system/multi-user.target.wants/ceph-radosgw.target → /lib/systemd/system/ceph-radosgw.target.
[ceph-node4] [DEBUG ] Created symlink /etc/systemd/system/ceph.target.wants/ceph-radosgw.target → /lib/systemd/system/ceph-radosgw.target.
[ceph-node4] [DEBUG ] Setting up ceph-osd (16.2.5-1bionic) ...
[ceph-node4] [DEBUG ] Created symlink /etc/systemd/system/multi-user.target.wants/ceph-osd.target → /lib/systemd/system/ceph-osd.target.
[ceph-node4] [DEBUG ] Created symlink /etc/systemd/system/ceph.target.wants/ceph-osd.target → /lib/systemd/system/ceph-osd.target.
[ceph-node4] [DEBUG ] Setting up ceph-mds (16.2.5-1bionic) ...
[ceph-node4] [DEBUG ] Created symlink /etc/systemd/system/multi-user.target.wants/ceph-mds.target → /lib/systemd/system/ceph-mds.target.
[ceph-node4] [DEBUG ] Created symlink /etc/systemd/system/ceph.target.wants/ceph-mds.target → /lib/systemd/system/ceph-mds.target.
[ceph-node4] [DEBUG ] Setting up python3-makie (1.0.7+ds1-1) ...
[ceph-node4] [DEBUG ] Setting up ceph-mon (16.2.5-1bionic) ...
[ceph-node4] [DEBUG ] Created symlink /etc/systemd/system/multi-user.target.wants/ceph-mon.target → /lib/systemd/system/ceph-mon.target.
[ceph-node4] [DEBUG ] Created symlink /etc/systemd/system/ceph.target.wants/ceph-mon.target → /lib/systemd/system/ceph-mon.target.
[ceph-node4] [DEBUG ] Setting up ceph-mgr-modules-core (16.2.5-1bionic) ...
[ceph-node4] [DEBUG ] Setting up python3-pastedeploy (1.5.2-4) ...
[ceph-node4] [DEBUG ] Setting up python3-websupport (2.0.28-1ubuntu1) ...
[ceph-node4] [DEBUG ] Setting up python3-pecan (1.2.1-2) ...
[ceph-node4] [DEBUG ] update-alternatives: using /usr/bin/python3-pecan to provide /usr/bin/pecan (pecan) in auto mode
[ceph-node4] [DEBUG ] update-alternatives: using /usr/bin/python3-gunicorn_pecan to provide /usr/bin/gunicorn_pecan (gunicorn_pecan) in auto mode
[ceph-node4] [DEBUG ] Setting up ceph-mgr (16.2.5-1bionic) ...
[ceph-node4] [DEBUG ] Created symlink /etc/systemd/system/multi-user.target.wants/ceph-mgr.target → /lib/systemd/system/ceph-mgr.target.
[ceph-node4] [DEBUG ] Created symlink /etc/systemd/system/ceph.target.wants/ceph-mgr.target → /lib/systemd/system/ceph-mgr.target.
[ceph-node4] [DEBUG ] Setting up ceph (16.2.5-1bionic) ...
[ceph-node4] [DEBUG ] Processing triggers for systemd (237-3ubuntu10.43) ...
[ceph-node4] [DEBUG ] Processing triggers for man-db (2.8.3-2ubuntu0.1) ...      杰哥的截图水印
[ceph-node4] [DEBUG ] Processing triggers for ureadahead (0.100.0-21) ...
[ceph-node4] [DEBUG ] Processing triggers for libc-bin (2.27-3ubuntu1.4) ...
[ceph-node4] [INFO ] Running command: sudo ceph --version
[ceph-node4] [DEBUG ] ceph version 16.2.5 (0883bdea7337b95e4b611c768c0279868462204a) pacific (stable)
ceph@ceph-deploy:~/ceph-cluster$
```

[ceph-node1][INFO] Running command: sudo ceph --version

```
[ceph-node1][DEBUG] ceph version 13.2.10  
(564bdc4ae87418a232fc901524470e1a0f76d641) mimic (stable)  
[ceph_deploy.install][DEBUG] Detecting platform for host ceph-node2 ...  
[ceph-node2][DEBUG] connection detected need for sudo  
[ceph-node2][DEBUG] connected to host: ceph-node2  
[ceph-node2][DEBUG] detect platform information from remote host  
[ceph-node2][DEBUG] detect machine type  
[ceph_deploy.install][INFO] Distro info: CentOS Linux 7.8.2003 Core  
[ceph-node2][INFO] installing Ceph on ceph-node2  
[ceph-node2][INFO] Running command: sudo yum clean all  
[ceph-node2][DEBUG] Loaded plugins: fastestmirror  
[ceph-node2][DEBUG] Cleaning repos: Ceph Ceph-noarch base ceph-source epel  
extras updates  
[ceph-node2][DEBUG] Cleaning up list of fastest mirrors  
[ceph-node2][INFO] Running command: sudo yum -y install ceph ceph-radosgw  
[ceph-node2][DEBUG] Loaded plugins: fastestmirror  
[ceph-node2][DEBUG] Determining fastest mirrors  
[ceph-node2][DEBUG] * base: mirrors.huaweicloud.com  
[ceph-node2][DEBUG] * extras: mirrors.bfsu.edu.cn  
[ceph-node2][DEBUG] * updates: mirrors.bfsu.edu.cn  
[ceph-node2][WARNIN]  
http://mirrors.nju.edu.cn/centos/7.9.2009/os/x86_64/repo/repodata/repomd.xml: [Errno 14]  
curl#7 - "Failed to connect to 2001:da8:1007:4011::3: Network is unreachable"  
[ceph-node2][WARNIN] Trying other mirror.  
[ceph-node2][WARNIN]  
https://mirrors.tuna.tsinghua.edu.cn/ceph/rpm-octopus/el7/x86_64/repo/repodata/35a2ee12  
2e0f37aa9b696946be043960d0e46ae2a766c6eac59614a075153288-primary.xml.gz:  
[Errno 14] curl#7 - "Failed to connect to 2402:f000:1:408:8100::1: Network is  
unreachable"  
[ceph-node2][WARNIN] Trying other mirror.  
[ceph-node2][DEBUG] Resolving Dependencies  
[ceph-node2][DEBUG] --> Running transaction check  
[ceph-node2][DEBUG] --> Package ceph.x86_64 2:13.2.10-0.el7 will be installed  
[ceph-node2][DEBUG] --> Processing Dependency: ceph-mds = 2:13.2.10-0.el7 for  
package: 2:ceph-13.2.10-0.el7.x86_64  
[ceph-node2][DEBUG] --> Processing Dependency: ceph-osd = 2:13.2.10-0.el7 for  
package: 2:ceph-13.2.10-0.el7.x86_64  
[ceph-node2][DEBUG] --> Processing Dependency: ceph-mon = 2:13.2.10-0.el7 for  
package: 2:ceph-13.2.10-0.el7.x86_64  
[ceph-node2][DEBUG] --> Processing Dependency: ceph-mgr = 2:13.2.10-0.el7 for  
package: 2:ceph-13.2.10-0.el7.x86_64  
[ceph-node2][DEBUG] --> Package ceph-radosgw.x86_64 2:13.2.10-0.el7 will be  
installed  
[ceph-node2][DEBUG] --> Processing Dependency: ceph-base = 2:13.2.10-0.el7 for
```

```
package: 2:ceph-radosgw-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: ceph-selinux = 2:13.2.10-0.el7 for
package: 2:ceph-radosgw-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: librsgw2 = 2:13.2.10-0.el7 for
package: 2:ceph-radosgw-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: librados2 = 2:13.2.10-0.el7 for
package: 2:ceph-radosgw-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: mailcap for package:
2:ceph-radosgw-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: librados.so.2()(64bit) for package:
2:ceph-radosgw-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: libtcmalloc.so.4()(64bit) for package:
2:ceph-radosgw-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: libibverbs.so.1()(64bit) for package:
2:ceph-radosgw-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: liboauth.so.0()(64bit) for package:
2:ceph-radosgw-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: libceph-common.so.0()(64bit) for
package: 2:ceph-radosgw-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Running transaction check
[ceph-node2][DEBUG ] ---> Package ceph-base.x86_64 2:13.2.10-0.el7 will be installed
[ceph-node2][DEBUG ] --> Processing Dependency: libcephfs2 = 2:13.2.10-0.el7 for
package: 2:ceph-base-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: librbd1 = 2:13.2.10-0.el7 for
package: 2:ceph-base-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: ceph-common = 2:13.2.10-0.el7 for
package: 2:ceph-base-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: gdisk for package:
2:ceph-base-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: python-setuptools for package:
2:ceph-base-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: psmisc for package:
2:ceph-base-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: cryptsetup for package:
2:ceph-base-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: python-requests for package:
2:ceph-base-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: libleveldb.so.1()(64bit) for package:
2:ceph-base-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: libfuse.so.2()(64bit) for package:
2:ceph-base-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: libltng-ust.so.0()(64bit) for package:
2:ceph-base-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] ---> Package ceph-mds.x86_64 2:13.2.10-0.el7 will be installed
```

```
[ceph-node2][DEBUG ] ---> Package ceph-mgr.x86_64 2:13.2.10-0.el7 will be installed
[ceph-node2][DEBUG ] --> Processing Dependency: python-cherrypy for package:
2:ceph-mgr-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: python-werkzeug for package:
2:ceph-mgr-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: python-six for package:
2:ceph-mgr-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: python-routes for package:
2:ceph-mgr-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: python-bcrypt for package:
2:ceph-mgr-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: python-pecan for package:
2:ceph-mgr-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: pyOpenSSL for package:
2:ceph-mgr-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: python-jinja2 for package:
2:ceph-mgr-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] ---> Package ceph-mon.x86_64 2:13.2.10-0.el7 will be installed
[ceph-node2][DEBUG ] ---> Package ceph-osd.x86_64 2:13.2.10-0.el7 will be installed
[ceph-node2][DEBUG ] ---> Package ceph-selinux.x86_64 2:13.2.10-0.el7 will be
installed
[ceph-node2][DEBUG ] ---> Package gperftools-libs.x86_64 0:2.6.1-1.el7 will be installed
[ceph-node2][DEBUG ] ---> Package libibverbs.x86_64 0:22.4-6.el7_9 will be installed
[ceph-node2][DEBUG ] --> Processing Dependency: rdma-core(x86-64) = 22.4-6.el7_9
for package: libibverbs-22.4-6.el7_9.x86_64
[ceph-node2][DEBUG ] ---> Package liboauth.x86_64 0:2.6.2-1.el7 will be installed
[ceph-node2][DEBUG ] ---> Package librados2.x86_64 2:13.2.10-0.el7 will be installed
[ceph-node2][DEBUG ] ---> Package librgw2.x86_64 2:13.2.10-0.el7 will be installed
[ceph-node2][DEBUG ] ---> Package mailcap.noarch 0:2.1.41-2.el7 will be installed
[ceph-node2][DEBUG ] --> Running transaction check
[ceph-node2][DEBUG ] ---> Package ceph-common.x86_64 2:13.2.10-0.el7 will be
installed
[ceph-node2][DEBUG ] --> Processing Dependency: python-cephfs = 2:13.2.10-0.el7 for
package: 2:ceph-common-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: libradosstriper1 = 2:13.2.10-0.el7
for package: 2:ceph-common-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: python-rgw = 2:13.2.10-0.el7 for
package: 2:ceph-common-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: python-rbd = 2:13.2.10-0.el7 for
package: 2:ceph-common-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: python-rados = 2:13.2.10-0.el7 for
package: 2:ceph-common-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: python-ceph-argparse =
2:13.2.10-0.el7 for package: 2:ceph-common-13.2.10-0.el7.x86_64
```

```
[ceph-node2][DEBUG ] --> Processing Dependency: python-prettytable for package
2:ceph-common-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: libbabeltrace-ctf.so.1()(64bit) for
package: 2:ceph-common-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: libbabeltrace.so.1()(64bit) for
package: 2:ceph-common-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: libradosstriper.so.1()(64bit) for
package: 2:ceph-common-13.2.10-0.el7.x86_64
[ceph-node2][DEBUG ] ---> Package cryptsetup.x86_64 0:2.0.3-6.el7 will be installed
[ceph-node2][DEBUG ] ---> Package fuse-libs.x86_64 0:2.9.2-11.el7 will be installed
[ceph-node2][DEBUG ] ---> Package gdisk.x86_64 0:0.8.10-3.el7 will be installed
[ceph-node2][DEBUG ] ---> Package leveldb.x86_64 0:1.12.0-11.el7 will be installed
[ceph-node2][DEBUG ] ---> Package libcephfs2.x86_64 2:13.2.10-0.el7 will be installed
[ceph-node2][DEBUG ] ---> Package librbd1.x86_64 2:13.2.10-0.el7 will be installed
[ceph-node2][DEBUG ] ---> Package lttng-ust.x86_64 0:2.4.1-4.el7 will be installed
[ceph-node2][DEBUG ] --> Processing Dependency: liburcu-bp.so.1()(64bit) for package:
lttng-ust-2.4.1-4.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: liburcu-cds.so.1()(64bit) for package:
lttng-ust-2.4.1-4.el7.x86_64
[ceph-node2][DEBUG ] ---> Package psmisc.x86_64 0:22.20-17.el7 will be installed
[ceph-node2][DEBUG ] ---> Package pyOpenSSL.x86_64 0:0.13.1-4.el7 will be installed
[ceph-node2][DEBUG ] ---> Package python-cherrypy.noarch 0:3.2.2-4.el7 will be
installed
[ceph-node2][DEBUG ] ---> Package python-jinja2.noarch 0:2.7.2-4.el7 will be installed
[ceph-node2][DEBUG ] --> Processing Dependency: python-babel >= 0.8 for package:
python-jinja2-2.7.2-4.el7.noarch
[ceph-node2][DEBUG ] --> Processing Dependency: python-markupsafe for package:
python-jinja2-2.7.2-4.el7.noarch
[ceph-node2][DEBUG ] ---> Package python-pecan.noarch 0:0.4.5-2.el7 will be installed
[ceph-node2][DEBUG ] --> Processing Dependency: python-mako >= 0.4.0 for package:
python-pecan-0.4.5-2.el7.noarch
[ceph-node2][DEBUG ] --> Processing Dependency: python-simplegeneric >= 0.8 for
package: python-pecan-0.4.5-2.el7.noarch
[ceph-node2][DEBUG ] --> Processing Dependency: python-webob >= 1.2 for package:
python-pecan-0.4.5-2.el7.noarch
[ceph-node2][DEBUG ] ---> Package python-webtest >= 1.3.1 for
package: python-pecan-0.4.5-2.el7.noarch
[ceph-node2][DEBUG ] --> Processing Dependency: python-singledispatch for package:
python-pecan-0.4.5-2.el7.noarch
[ceph-node2][DEBUG ] ---> Package python-requests.noarch 0:2.6.0-10.el7 will be
installed
[ceph-node2][DEBUG ] --> Processing Dependency: python-urllib3 >= 1.10.2-1 for
package: python-requests-2.6.0-10.el7.noarch
[ceph-node2][DEBUG ] --> Processing Dependency: python-chardet >= 2.2.1-1 for
```

```
package: python-requests-2.6.0-10.el7.noarch
[ceph-node2][DEBUG ] ---> Package python-routes.noarch 0:1.13-2.el7 will be installed
[ceph-node2][DEBUG ] --> Processing Dependency: python-repoze-lru for package:
python-routes-1.13-2.el7.noarch
[ceph-node2][DEBUG ] ---> Package python-setuptools.noarch 0:0.9.8-7.el7 will be
installed
[ceph-node2][DEBUG ] ] --> Processing Dependency: python-backports-ssl_match_hostname
for package: python-setuptools-0.9.8-7.el7.noarch
[ceph-node2][DEBUG ] ---> Package python-six.noarch 0:1.9.0-2.el7 will be installed
[ceph-node2][DEBUG ] ---> Package python-werkzeug.noarch 0:0.9.1-2.el7 will be
installed
[ceph-node2][DEBUG ] ---> Package python2-bcrypt.x86_64 0:3.1.6-2.el7 will be
installed
[ceph-node2][DEBUG ] --> Processing Dependency: python-cffi for package:
python2-bcrypt-3.1.6-2.el7.x86_64
[ceph-node2][DEBUG ] --> Processing Dependency: python2-six for package:
python2-bcrypt-3.1.6-2.el7.x86_64
[ceph-node2][DEBUG ] ---> Package rdma-core.x86_64 0:22.4-6.el7_9 will be installed
[ceph-node2][DEBUG ] --> Processing Dependency: pciutils for package:
rdma-core-22.4-6.el7_9.x86_64
[ceph-node2][DEBUG ] --> Running transaction check
[ceph-node2][DEBUG ] ---> Package libbabeltrace.x86_64 0:1.2.4-3.el7 will be installed
[ceph-node2][DEBUG ] ---> Package libradosstriper1.x86_64 2:13.2.10-0.el7 will be
installed
[ceph-node2][DEBUG ] ---> Package pciutils.x86_64 0:3.5.1-3.el7 will be installed
[ceph-node2][DEBUG ] ---> Package python-babel.noarch 0:0.9.6-8.el7 will be installed
[ceph-node2][DEBUG ] ---> Package python-backports-ssl_match_hostname.noarch
0:3.5.0.1-1.el7 will be installed
[ceph-node2][DEBUG ] --> Processing Dependency: python-ipaddress for package:
python-backports-ssl_match_hostname-3.5.0.1-1.el7.noarch
[ceph-node2][DEBUG ] --> Processing Dependency: python-backports for package:
python-backports-ssl_match_hostname-3.5.0.1-1.el7.noarch
[ceph-node2][DEBUG ] ---> Package python-ceph argparse.x86_64 2:13.2.10-0.el7 will
be installed
[ceph-node2][DEBUG ] ---> Package python-cephfs.x86_64 2:13.2.10-0.el7 will be
installed
[ceph-node2][DEBUG ] ---> Package python-cffi.x86_64 0:1.6.0-5.el7 will be installed
[ceph-node2][DEBUG ] --> Processing Dependency: python-pycparser for package:
python-cffi-1.6.0-5.el7.x86_64
[ceph-node2][DEBUG ] ---> Package python-chardet.noarch 0:2.2.1-3.el7 will be
installed
[ceph-node2][DEBUG ] ---> Package python-mako.noarch 0:0.8.1-2.el7 will be installed
[ceph-node2][DEBUG ] --> Processing Dependency: python-beaker for package:
```

```
python-mako-0.8.1-2.el7.noarch
[ceph-node2][DEBUG ] ---> Package python-markupsafe.x86_64 0:0.11-10.el7 will be
installed
[ceph-node2][DEBUG ] ---> Package python-prettytable.noarch 0:0.7.2-3.el7 will be
installed
[ceph-node2][DEBUG ] ---> Package python-rados.x86_64 2:13.2.10-0.el7 will be
installed
[ceph-node2][DEBUG ] ---> Package python-rbd.x86_64 2:13.2.10-0.el7 will be installed
[ceph-node2][DEBUG ] ---> Package python-repoze-lru.noarch 0:0.4-3.el7 will be
installed
[ceph-node2][DEBUG ] ---> Package python-rgw.x86_64 2:13.2.10-0.el7 will be installed
[ceph-node2][DEBUG ] ---> Package python-simplegeneric.noarch 0:0.8-7.el7 will be
installed
[ceph-node2][DEBUG ] ---> Package python-singledispatch.noarch 0:3.4.0.2-2.el7 will
be installed
[ceph-node2][DEBUG ] ---> Package python-urllib3.noarch 0:1.10.2-7.el7 will be
installed
[ceph-node2][DEBUG ] ---> Package python-webob.noarch 0:1.2.3-7.el7 will be installed
[ceph-node2][DEBUG ] ---> Package python-webtest.noarch 0:1.3.4-6.el7 will be
installed
[ceph-node2][DEBUG ] ---> Package python2-six.noarch 0:1.9.0-0.el7 will be installed
[ceph-node2][DEBUG ] ---> Package userspace-rcu.x86_64 0:0.7.16-1.el7 will be
installed
[ceph-node2][DEBUG ] --> Running transaction check
[ceph-node2][DEBUG ] ---> Package python-backports.x86_64 0:1.0-8.el7 will be
installed
[ceph-node2][DEBUG ] ---> Package python-beaker.noarch 0:1.5.4-10.el7 will be
installed
[ceph-node2][DEBUG ] --> Processing Dependency: python-paste for package:
python-beaker-1.5.4-10.el7.noarch
[ceph-node2][DEBUG ] ---> Package python-ipaddress.noarch 0:1.0.16-2.el7 will be
installed
[ceph-node2][DEBUG ] ---> Package python-pycparser.noarch 0:2.14-1.el7 will be
installed
[ceph-node2][DEBUG ] --> Processing Dependency: python-ply for package:
python-pycparser-2.14-1.el7.noarch
[ceph-node2][DEBUG ] --> Running transaction check
[ceph-node2][DEBUG ] ---> Package python-paste.noarch
0:1.7.5.1-9.20111221hg1498.el7 will be installed
[ceph-node2][DEBUG ] --> Processing Dependency: python-tempita for package:
python-paste-1.7.5.1-9.20111221hg1498.el7.noarch
[ceph-node2][DEBUG ] ---> Package python-ply.noarch 0:3.4-11.el7 will be installed
[ceph-node2][DEBUG ] --> Running transaction check
[ceph-node2][DEBUG ] ---> Package python-tempita.noarch 0:0.5.1-6.el7 will be
```

```

installed
[ceph-node2][DEBUG ] --> Finished Dependency Resolution
[ceph-node2][DEBUG ]
[ceph-node2][DEBUG ] Dependencies Resolved
[ceph-node2][DEBUG ]
[ceph-node2][DEBUG ]                                     ]
=====
=====
[ceph-node2][DEBUG ]  Package                               Arch      Version
Repository
[ceph-node2][DEBUG ]                                     ]]
Size
[ceph-node2][DEBUG ]                                     ]]
=====
=====
[ceph-node2][DEBUG ] Installing:
[ceph-node2][DEBUG ]  ceph                           x86_64  2:13.2.10-0.el7
Ceph    3.0 k
[ceph-node2][DEBUG ]  ceph-radosgw                  x86_64  2:13.2.10-0.el7
Ceph    4.5 M
[ceph-node2][DEBUG ] Installing for dependencies:
[ceph-node2][DEBUG ]  ceph-base                     x86_64  2:13.2.10-0.el7
Ceph    4.8 M
[ceph-node2][DEBUG ]  ceph-common                  x86_64  2:13.2.10-0.el7
Ceph    14 M
[ceph-node2][DEBUG ]  ceph-mds                      x86_64  2:13.2.10-0.el7
Ceph    1.7 M
[ceph-node2][DEBUG ]  ceph-mgr                      x86_64  2:13.2.10-0.el7
Ceph    3.5 M
[ceph-node2][DEBUG ]  ceph-mon                      x86_64  2:13.2.10-0.el7
Ceph    3.9 M
[ceph-node2][DEBUG ]  ceph-osd                      x86_64  2:13.2.10-0.el7
Ceph    13 M
[ceph-node2][DEBUG ]  ceph-selinux                 x86_64  2:13.2.10-0.el7
Ceph    21 k
[ceph-node2][DEBUG ]  cryptsetup                   x86_64  2.0.3-6.el7
base   154 k
[ceph-node2][DEBUG ]  fuse-libs                    x86_64  2.9.2-11.el7
base   93 k
[ceph-node2][DEBUG ]  gdisk                       x86_64  0.8.10-3.el7
base   190 k
[ceph-node2][DEBUG ]  gperftools-libs            x86_64  2.6.1-1.el7
base   272 k
[ceph-node2][DEBUG ]  leveldb                     x86_64  1.12.0-11.el7

```

```

epel    161 k
[ceph-node2][DEBUG ]    libbabeltrace                         x86_64 1.2.4-3.el7
epel    147 k
[ceph-node2][DEBUG ]    libcephfs2                           x86_64 2:13.2.10-0.el7
Ceph    434 k
[ceph-node2][DEBUG ]    libibverbs                           x86_64 22.4-6.el7_9
updates 269 k
[ceph-node2][DEBUG ]    liboath                             x86_64 2.6.2-1.el7
epel    51 k
[ceph-node2][DEBUG ]    librados2                           x86_64 2:13.2.10-0.el7
Ceph    2.9 M
[ceph-node2][DEBUG ]    libradosstriper1                     x86_64 2:13.2.10-0.el7
Ceph    329 k
[ceph-node2][DEBUG ]    librbd1                            x86_64 2:13.2.10-0.el7
Ceph    1.2 M
[ceph-node2][DEBUG ]    librgw2                            x86_64 2:13.2.10-0.el7
Ceph    2.0 M
[ceph-node2][DEBUG ]    ltntg-ust                           x86_64 2.4.1-4.el7
epel    176 k
[ceph-node2][DEBUG ]    mailcap                           noarch 2.1.41-2.el7
base    31 k
[ceph-node2][DEBUG ]    pciutils                           x86_64 3.5.1-3.el7
base    93 k
[ceph-node2][DEBUG ]    psmisc                            x86_64 22.20-17.el7
base    141 k
[ceph-node2][DEBUG ]    pyOpenSSL                           x86_64 0.13.1-4.el7
base    135 k
[ceph-node2][DEBUG ]    python-babel                          noarch 0.9.6-8.el7
base    1.4 M
[ceph-node2][DEBUG ]    python-backports                     x86_64 1.0-8.el7
base    5.8 k

```

此过程会在指定的 ceph node 节点按照串行的方式逐个服务器安装 epel 源和 ceph 源并按安装 ceph ceph-radosgw

```

[ceph-node1][DEBUG] Resolving Dependencies
[ceph-node1][DEBUG] --> Running transaction check
[ceph-node1][DEBUG] --> Package ceph.x86_64 2:13.2.10-0.el7 will be installed
[ceph-node1][DEBUG] --> Processing Dependency: ceph-mds = 2:13.2.10-0.el7 for package: 2:ceph-13.2.10-0.el7.x86_64
[ceph-node1][DEBUG] --> Processing Dependency: ceph-osd = 2:13.2.10-0.el7 for package: 2:ceph-13.2.10-0.el7.x86_64
[ceph-node1][DEBUG] --> Processing Dependency: ceph-mon = 2:13.2.10-0.el7 for package: 2:ceph-13.2.10-0.el7.x86_64
[ceph-node1][DEBUG] --> Processing Dependency: ceph-mgr = 2:13.2.10-0.el7 for package: 2:ceph-13.2.10-0.el7.x86_64
[ceph-node1][DEBUG] --> Package ceph-radosgw.x86_64 2:13.2.10-0.el7 will be installed
[ceph-node1][DEBUG] --> Processing Dependency: ceph-base = 2:13.2.10-0.el7 for package: 2:ceph-radosgw-13.2.10-0.el7.x86_64
[ceph-node1][DEBUG] --> Processing Dependency: ceph-selinux = 2:13.2.10-0.el7 for package: 2:ceph-radosgw-13.2.10-0.el7.x86_64
[ceph-node1][DEBUG] --> Processing Dependency: librgw2 = 2:13.2.10-0.el7 for package: 2:ceph-radosgw-13.2.10-0.el7.x86_64
[ceph-node1][DEBUG] --> Processing Dependency: librados2 = 2:13.2.10-0.el7 for package: 2:ceph-radosgw-13.2.10-0.el7.x86_64
[ceph-node1][DEBUG] --> Processing Dependency: mailcap for package: 2:ceph-radosgw-13.2.10-0.el7.x86_64
[ceph-node1][DEBUG] --> Processing Dependency: librados.so.2() (64bit) for package: 2:ceph-radosgw-13.2.10-0.el7.x86_64
[ceph-node1][DEBUG] --> Processing Dependency: libtcmalloc.so.4() (64bit) for package: 2:ceph-radosgw-13.2.10-0.el7.x86_64
[ceph-node1][DEBUG] --> Processing Dependency: libibverbs.so.1() (64bit) for package: 2:ceph-radosgw-13.2.10-0.el7.x86_64
[ceph-node1][DEBUG] --> Processing Dependency: liboath.so.0() (64bit) for package: 2:ceph-radosgw-13.2.10-0.el7.x86_64
[ceph-node1][DEBUG] --> Processing Dependency: libceph-common.so.0() (64bit) for package: 2:ceph-radosgw-13.2.10-0.el7.x86_64
[ceph-node1][DEBUG] --> Running transaction check
[ceph-node1][DEBUG] --> Package ceph-base.x86_64 2:13.2.10-0.el7 will be installed
[ceph-node1][DEBUG] --> Processing Dependency: libcephfs2 = 2:13.2.10-0.el7 for package: 2:ceph-base-13.2.10-0.el7.x86_64
[ceph-node1][DEBUG] --> Processing Dependency: librbd1 = 2:13.2.10-0.el7 for package: 2:ceph-base-13.2.10-0.el7.x86_64
[ceph-node1][DEBUG] --> Processing Dependency: ceph-common = 2:13.2.10-0.el7 for package: 2:ceph-base-13.2.10-0.el7.x86_64
[ceph-node1][DEBUG] --> Processing Dependency: gdisk for package: 2:ceph-base-13.2.10-0.el7.x86_64
[ceph-node1][DEBUG] --> Processing Dependency: python-setuptools for package: 2:ceph-base-13.2.10-0.el7.x86_64
[ceph-node1][DEBUG] --> Processing Dependency: psmisc for package: 2:ceph-base-13.2.10-0.el7.x86_64
[ceph-node1][DEBUG] --> Processing Dependency: cryptsetup for package: 2:ceph-base-13.2.10-0.el7.x86_64
[ceph-node1][DEBUG] --> Processing Dependency: python-requests for package: 2:ceph-base-13.2.10-0.el7.x86_64
[ceph-node1][DEBUG] --> Processing Dependency: libleveldb.so.1() (64bit) for package: 2:ceph-base-13.2.10-0.el7.x86_64
[ceph-node1][DEBUG] --> Processing Dependency: libfuse.so.2() (64bit) for package: 2:ceph-base-13.2.10-0.el7.x86_64
[ceph-node1][DEBUG] --> Processing Dependency: liblbtng-ust.so.0() (64bit) for package: 2:ceph-base-13.2.10-0.el7.x86_64
[ceph-node1][DEBUG] --> Package ceph-mds.x86_64 2:13.2.10-0.el7 will be installed
[ceph-node1][DEBUG] --> Package ceph-mgr.x86_64 2:13.2.10-0.el7 will be installed
[ceph-node1][DEBUG] --> Processing Dependency: python-cherrypy for package: 2:ceph-mgr-13.2.10-0.el7.x86_64
[ceph-node1][DEBUG] --> Processing Dependency: python-werkzeug for package: 2:ceph-mgr-13.2.10-0.el7.x86_64

```

杰哥的截图水印

```

[ceph-node2][DEBUG] python-werkzeug.noarch 0:0.9.1-2.el7
[ceph-node2][DEBUG] python2-bcrypt.x86_64 0:3.1.6-2.el7
[ceph-node2][DEBUG] python2-six.noarch 0:1.9.0-0.el7
[ceph-node2][DEBUG] rdma-core.x86_64 0:22.4.6.el7_9
[ceph-node2][DEBUG] userspace-rcu.x86_64 0:0.7.16-1.el7
[ceph-node2][DEBUG]
[ceph-node2][DEBUG] Complete!
[ceph-node2][INFO] Running command: sudo ceph --version
[ceph-node2][DEBUG] ceph version 13.2.10 (564bdc4ae87418a232fc901524470e1a0f76d641) mimic (stable)
[ceph_deploy.install][DEBUG] Detecting platform for host ceph-node3 ...
The authenticity of host 'ceph-node3 (172.31.6.103)' can't be established.
ECDSA key fingerprint is SHA256:mTkWCenFt66kYek1uMN0we0z8c1X0yDMluZFEwR62pE.
ECDSA key fingerprint is MD5:16:de:cfa:3:96:c4:bd:ea:45:34:22:59:d2:39:c2:bd.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ceph-node3' (ECDSA) to the list of known hosts.
[ceph-node3][DEBUG] connection detected need for sudo
[ceph-node3][DEBUG] connected to host: ceph-node3
[ceph-node3][DEBUG] detect platform information from remote host
[ceph-node3][DEBUG] detect machine type
[ceph_deploy.install][INFO] Distro info: CentOS Linux 7.8.2003 Core
[ceph-node3][INFO] installing Ceph on ceph-node3
[ceph-node3][INFO] Running command: sudo yum clean all
[ceph-node3][DEBUG] Loaded plugins: fastestmirror
[ceph-node3][DEBUG] Cleaning repos: Ceph Ceph-noarch base ceph-source epel extras updates
[ceph-node3][DEBUG] Cleaning up list of fastest mirrors
[ceph-node3][INFO] Running command: sudo yum -y install ceph ceph-radosgw
[ceph-node3][DEBUG] Loaded plugins: fastestmirror
[ceph-node3][DEBUG] Determining fastest mirrors
[ceph-node3][DEBUG] * base: mirrors.tuna.tsinghua.edu.cn
[ceph-node3][DEBUG] * extras: mirrors.tuna.tsinghua.edu.cn
[ceph-node3][DEBUG] * updates: mirrors.tuna.tsinghua.edu.cn

```

杰哥的截图水印

初始化完成

```

[ceph-node3][DEBUG] python-pecan.noarch 0:0.4.5-2.el7
[ceph-node3][DEBUG] python-ply.noarch 0:3.4-11.el7
[ceph-node3][DEBUG] python-prettytable.noarch 0:0.7.2-3.el7
[ceph-node3][DEBUG] python-pycparser.noarch 0:2.14-1.el7
[ceph-node3][DEBUG] python-rados.x86_64 2:13.2.10-0.el7
[ceph-node3][DEBUG] python-rbd.x86_64 2:13.2.10-0.el7
[ceph-node3][DEBUG] python-repoze-lru.noarch 0:0.4-3.el7
[ceph-node3][DEBUG] python-requests.noarch 0:2.6.0-10.el7
[ceph-node3][DEBUG] python-rgw.x86_64 2:13.2.10-0.el7
[ceph-node3][DEBUG] python-routes.noarch 0:1.13-2.el7
[ceph-node3][DEBUG] python-setuptools.noarch 0:0.9.8-7.el7
[ceph-node3][DEBUG] python-simplegeneric.noarch 0:0.8-7.el7
[ceph-node3][DEBUG] python-singledispatch.noarch 0:3.4.0.2-2.el7
[ceph-node3][DEBUG] python-six.noarch 0:1.9.0-2.el7
[ceph-node3][DEBUG] python-tempita.noarch 0:0.5.1-6.el7
[ceph-node3][DEBUG] python-urllib3.noarch 0:1.10.2-7.el7
[ceph-node3][DEBUG] python-webob.noarch 0:1.2.3-7.el7
[ceph-node3][DEBUG] python-webtest.noarch 0:1.3.4-6.el7
[ceph-node3][DEBUG] python-werkzeug.noarch 0:0.9.1-2.el7
[ceph-node3][DEBUG] python2-bcrypt.x86_64 0:3.1.6-2.el7
[ceph-node3][DEBUG] python2-six.noarch 0:1.9.0-0.el7
[ceph-node3][DEBUG] rdma-core.x86_64 0:22.4-6.el7_9
[ceph-node3][DEBUG] userspace-rcu.x86_64 0:0.7.16-1.el7
[ceph-node3][DEBUG]
[ceph-node3][DEBUG] Complete!
[ceph-node3][INFO] Running command: sudo ceph --version
[ceph-node3][DEBUG] ceph version 13.2.10 (564bdc4ae87418a232fc901524470e1a0f76d641) mimic (stable)
[ceph@ceph-deploy ceph-cluster]$

```

杰哥的截图水印

3.4.7:配置 mon 节点并生成及同步秘钥:

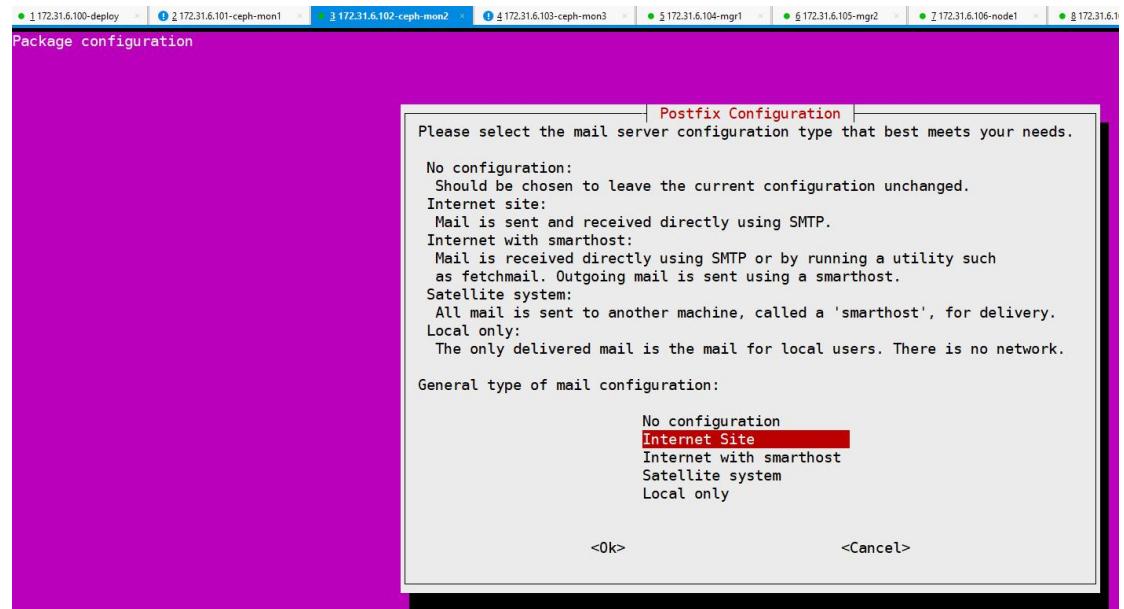
在各 mon 节点按照组件 ceph-mon, 并通过初始化 mon 节点, mon 节点 ha 还可以后期横向扩容。

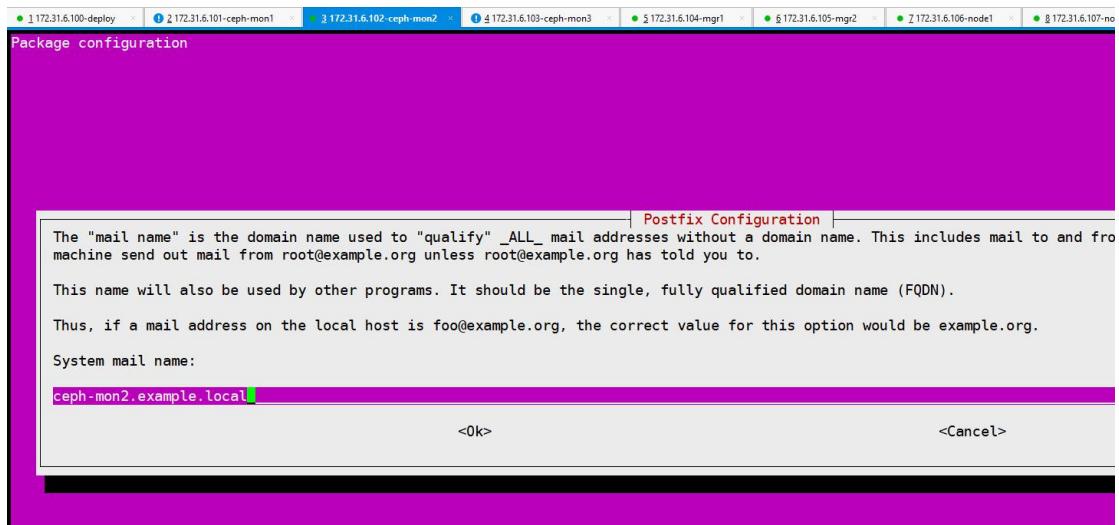
Ubuntu 安装 ceph-mon:

```

root@ceph-mon1:~# apt install ceph-mon
root@ceph-mon2:~# apt install ceph-mon
root@ceph-mon3:~# apt install ceph-mon

```





Centos 安裝 ceph-mon:::

```
[root@ceph-mon1 ~]# yum install ceph-mon -y

[ceph@ceph-deploy ceph-cluster]$ pwd
/home/ceph/ceph-cluster

[ceph@ceph-deploy ceph-cluster]$ ceph-deploy mon create-initial
ceph@ceph-deploy:~/ceph-cluster$ ceph-deploy mon create-initial
[ceph_deploy.conf][DEBUG ] found configuration file at: /home/ceph/.cephdeploy.conf
[ceph_deploy.cli][INFO  ] Invoked (2.0.1): /usr/bin/ceph-deploy mon create-initial
[ceph_deploy.cli][INFO  ] ceph-deploy options:
[ceph_deploy.cli][INFO  ]   username           : None
[ceph_deploy.cli][INFO  ]   verbose            : False
[ceph_deploy.cli][INFO  ]   overwrite_conf     : False
[ceph_deploy.cli][INFO  ]   subcommand          : create-initial
[ceph_deploy.cli][INFO  ]   quiet              : False
[ceph_deploy.cli][INFO  ]   cd_conf             : <ceph_deploy.conf.cephdeploy.Conf instance at 0x7f46aaf79fa0>
[ceph_deploy.cli][INFO  ]   cluster             : ceph
[ceph_deploy.cli][INFO  ]   func                : <function mon at 0x7f46aaf5cad0>
[ceph_deploy.cli][INFO  ]   ceph_conf          : None
[ceph_deploy.cli][INFO  ]   keyrings           : None
[ceph_deploy.cli][INFO  ]   default_release    : False
[ceph_deploy.mon][DEBUG ] Deploying mon, cluster ceph hosts ceph-mon1
[ceph_deploy.mon][DEBUG ] detecting platform for host ceph-mon1 ...
The authenticity of host 'ceph-mon1 (172.31.6.101)' can't be established.
ECDSA key fingerprint is SHA256:9khYc5k68pyboHx6VtTk2Id+y5UEBN3P0ZyM0srTZBc.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ceph-mon1' (ECDSA) to the list of known hosts.
ceph@ceph-mon1's password:
[ceph-mon1][DEBUG ] connection detected need for sudo
ceph@ceph-mon1's password:
[ceph-mon1][DEBUG ] connected to host: ceph-mon1
[ceph-mon1][DEBUG ] detect platform information from remote host
[ceph-mon1][DEBUG ] detect machine type
[ceph-mon1][DEBUG ] find the location of an executable
[ceph_deploy.mon][INFO  ] distro info: Ubuntu 18.04 bionic
[ceph-mon1][DEBUG ] determining if provided host has same hostname in remote
[ceph-mon1][DEBUG ] get remote short hostname
[ceph-mon1][DEBUG ] deploying mon to ceph-mon1
[ceph-mon1][DEBUG ] get remote short hostname
[ceph-mon1][DEBUG ] remote hostname: ceph-mon1
[ceph-mon1][DEBUG ] write cluster configuration to /etc/ceph/{cluster}.conf
```

```
[ceph-mon1] [DEBUG ] connected to host: ceph-mon1
[ceph-mon1] [DEBUG ] detect platform information from remote host
[ceph-mon1] [DEBUG ] detect machine type
[ceph-mon1] [DEBUG ] get remote short hostname
[ceph-mon1] [DEBUG ] fetch remote file
[ceph-mon1] [INFO  ] Running command: sudo /usr/bin/ceph --connect-timeout=25 --cluster=ceph
[ceph-mon1] [INFO  ] Running command: sudo /usr/bin/ceph --connect-timeout=25 --cluster=ceph
min
[ceph-mon1] [INFO  ] Running command: sudo /usr/bin/ceph --connect-timeout=25 --cluster=ceph
otstrap-mds
[ceph-mon1] [INFO  ] Running command: sudo /usr/bin/ceph --connect-timeout=25 --cluster=ceph
otstrap-mgr
[ceph-mon1] [INFO  ] Running command: sudo /usr/bin/ceph --connect-timeout=25 --cluster=ceph
otstrap-osd
[ceph-mon1] [INFO  ] Running command: sudo /usr/bin/ceph --connect-timeout=25 --cluster=ceph
otstrap-rgw
[ceph_deploy.gatherkeys] [INFO  ] Storing ceph.client.admin.keyring
[ceph_deploy.gatherkeys] [INFO  ] Storing ceph.bootstrap-mds.keyring
[ceph_deploy.gatherkeys] [INFO  ] Storing ceph.bootstrap-mgr.keyring
[ceph_deploy.gatherkeys] [INFO  ] keyring 'ceph.mon.keyring' already exists
[ceph_deploy.gatherkeys] [INFO  ] Storing ceph.bootstrap-osd.keyring
[ceph_deploy.gatherkeys] [INFO  ] Storing ceph.bootstrap-rgw.keyring
[ceph_deploy.gatherkeys] [INFO  ] Destroy temp directory /tmp/tmpcDxlEf
ceph@ceph-deploy:~/ceph-cluster$
```

3.4.8:验证 mon 节点:

验证在 mon 定节点已经自动安装并启动了 ceph-mon 服务，并且后期在 ceph-deploy 节点初始化目录会生成一些 bootstrap ceph mds/mgr/osd/rgw 等服务的 keyring 认证文件，这些初始化文件拥有对 ceph 集群的最高权限，所以一定要保存好。

```
[root@ceph-mon1 ~]# ps -ef | grep ceph-mon
ceph      4526      1  0 16:47 ?        00:00:00 /usr/bin/ceph-mon -f --cluster
ceph --id ceph-mon1 --setuser ceph --setgroup ceph
```

3.4.9:分发 admin 秘钥:

在 ceph-deploy 节点把配置文件和 admin 密钥拷贝至 Ceph 集群需要执行 ceph 管理命令的节点，从而不需要后期通过 ceph 命令对 ceph 集群进行管理配置的时候每次都需要指定 ceph-mon 节点地址和 ceph.client.admin.keyring 文件，另外各 ceph-mon 节点也需要同步 ceph 的集群配置文件与认证文件。

如果在 ceph-deploy 节点管理集群：

```
root@ceph-deploy:~# apt install ceph-common #先安装 ceph 的公共组件
root@ceph-node1:~# apt install ceph-common -y
root@ceph-node2:~# apt install ceph-common -y
root@ceph-node3:~# apt install ceph-common -y
root@ceph-node4:~# apt install ceph-common -y
```

```
ceph@ceph-deploy:/home/ceph/ceph-cluster$ ceph-deploy admin ceph-node1
ceph-node2 ceph-node3 ceph-node4
```

```
ceph@ceph-deploy:/home/ceph/ceph-cluster$ ceph-deploy admin ceph-node1 ceph-node2 ceph-node3 ceph-node4
[ceph_deploy.conf] [DEBUG] found configuration file at: /var/lib/ceph/.cephdeploy.conf
[ceph_deploy.cli] [INFO] Invoked (2.0.1): /usr/bin/ceph-deploy admin ceph-node1 ceph-node2 ceph-node3 ceph-node4
[ceph_deploy.cli] [INFO] ceph-deploy options:
[ceph_deploy.cli] [INFO]   username : None
[ceph_deploy.cli] [INFO]   verbose : False
[ceph_deploy.cli] [INFO]   overwrite_conf : False
[ceph_deploy.cli] [INFO]   quiet : False
[ceph_deploy.cli] [INFO]   cd_conf : <ceph_deploy.conf.cephdeploy.Conf instance at 0x7fe7968b0140>
[ceph_deploy.cli] [INFO]   cluster : ceph
[ceph_deploy.cli] [INFO]   client : ['ceph-node1', 'ceph-node2', 'ceph-node3', 'ceph-node4']
[ceph_deploy.cli] [INFO]   func : <function admin at 0x7fe7971b1a50>
[ceph_deploy.cli] [INFO]   ceph_conf : None
[ceph_deploy.cli] [INFO]   default_release : False
[ceph_deploy.admin] [DEBUG] Pushing admin keys and conf to ceph-node1
[ceph-node1] [DEBUG] connection detected need for sudo
[ceph-node1] [DEBUG] connected to host: ceph-node1
[ceph-node1] [DEBUG] detect platform information from remote host
[ceph-node1] [DEBUG] detect machine type
[ceph-node1] [DEBUG] write cluster configuration to /etc/ceph/{cluster}.conf
[ceph_deploy.admin] [DEBUG] Pushing admin keys and conf to ceph-node2
[ceph-node2] [DEBUG] connection detected need for sudo
[ceph-node2] [DEBUG] connected to host: ceph-node2
[ceph-node2] [DEBUG] detect platform information from remote host
[ceph-node2] [DEBUG] detect machine type
[ceph-node2] [DEBUG] write cluster configuration to /etc/ceph/{cluster}.conf
[ceph_deploy.admin] [DEBUG] Pushing admin keys and conf to ceph-node3
[ceph-node3] [DEBUG] connection detected need for sudo
```

3.4.10:ceph 节点验证秘钥:

到 node 节点验证 key 文件

```
[ceph@ceph-node2 ~]$ ll /etc/ceph/
total 12
-rw----- 1 root root 151 Oct 21 17:01 ceph.client.admin.keyring
-rw-r--r-- 1 root root 263 Oct 21 17:01 ceph.conf
-rw-r--r-- 1 root root  92 Apr 24 01:07 rbdmap
-rw----- 1 root root     0 Oct 21 17:01 tmpoJElzCg
```

认证文件的属主和属组为了安全考虑， 默认设置为了 root 用户和 root 组， 如果需要 ceph 用户也能执行 ceph 命令， 那么就需要对 ceph 用户进行授权，

```
root@ceph-node1:~# setfacl -m u:ceph:rw /etc/ceph/ceph.client.admin.keyring
root@ceph-node2:~# setfacl -m u:ceph:rw /etc/ceph/ceph.client.admin.keyring
root@ceph-node3:~# setfacl -m u:ceph:rw /etc/ceph/ceph.client.admin.keyring
root@ceph-node4:~# setfacl -m u:ceph:rw /etc/ceph/ceph.client.admin.keyring
```

3.4.11:配置 manager 节点:

ceph 的 Luminous 及以上版本有 manager 节点，早期的版本没有。

3.4.12:部署 ceph-mgr 节点:

mgr 节点需要读取 ceph 的配置文件，即/etc/ceph 目录中的配置文件

```
[ceph@ceph-deploy ceph-cluster]$ ceph-deploy mgr --help
usage: ceph-deploy mgr [-h] {create} ...
```

Ceph MGR daemon management

positional arguments:

```

{create}
    create      Deploy Ceph MGR on remote host(s)

optional arguments:
  -h, --help  show this help message and exit

#初始化 ceph-mgr 节点:
[root@ceph-mgr1 ~]#apt install ceph-mgr

[root@ceph-mgr1 ~]# yum install ceph-mgr

[ceph@ceph-deploy ceph-cluster]$ ceph-deploy  mgr create ceph-mgr1
ceph@ceph-deploy:/home/ceph/ceph-cluster$ ceph-deploy  mgr create ceph-mgr1
[ceph_deploy.conf][DEBUG ] found configuration file at: /var/lib/ceph/.cephdeploy.conf
[ceph_deploy.cli][INFO  ] Invoked (2.0.1): /usr/bin/ceph-deploy mgr create ceph-mgr1
[ceph_deploy.cli][INFO  ] ceph-deploy options:
[ceph_deploy.cli][INFO  ]   username                  : None
[ceph_deploy.cli][INFO  ]   verbose                   : False
[ceph_deploy.cli][INFO  ]   mgr                      : [('ceph-mgr1',
'ceph-mgr1')]
[ceph_deploy.cli][INFO  ]   overwrite_conf           : False
[ceph_deploy.cli][INFO  ]   subcommand                : create
[ceph_deploy.cli][INFO  ]   quiet                     : False
[ceph_deploy.cli][INFO  ]   cd_conf                   :
<ceph_deploy.conf.cephdeploy.Conf instance at 0x7f9fc5106be0>
[ceph_deploy.cli][INFO  ]   cluster                   : ceph
[ceph_deploy.cli][INFO  ]   func                     : <function mgr at
0x7f9fc5566150>
[ceph_deploy.cli][INFO  ]   ceph_conf                : None
[ceph_deploy.cli][INFO  ]   default_release          : False
[ceph_deploy.mgr][DEBUG ] Deploying mgr, cluster ceph hosts ceph-mgr1:ceph-mgr1
The authenticity of host 'ceph-mgr1 (172.31.6.104)' can't be established.
ECDSA               key               fingerprint           is
SHA256:9kHyC5k68pyboHx6VtTk2Id+y5UEBN3P0ZyM0srTZBc.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ceph-mgr1,172.31.6.104' (ECDSA) to the list of known
hosts.
ceph@ceph-mgr1's password:
[ceph-mgr1][DEBUG ] connection detected need for sudo
ceph@ceph-mgr1's password:
[ceph-mgr1][DEBUG ] connected to host: ceph-mgr1
[ceph-mgr1][DEBUG ] detect platform information from remote host
[ceph-mgr1][DEBUG ] detect machine type
[ceph_deploy.mgr][INFO  ] Distro info: Ubuntu 18.04 bionic

```

```
[ceph_deploy.mgr][DEBUG ] remote host will use systemd
[ceph_deploy.mgr][DEBUG ] deploying mgr bootstrap to ceph-mgr1
[ceph-mgr1][DEBUG ] write cluster configuration to /etc/ceph/{cluster}.conf
[ceph-mgr1][WARNIN] mgr keyring does not exist yet, creating one
[ceph-mgr1][DEBUG ] create a keyring file
[ceph-mgr1][DEBUG ] create path recursively if it doesn't exist
[ceph-mgr1][INFO    ] Running command: sudo ceph --cluster ceph --name
client.bootstrap-mgr --keyring /var/lib/ceph/bootstrap-mgr/ceph.keyring auth
get-or-create mgr.ceph-mgr1 mon allow profile mgr osd allow * mds allow * -o
/var/lib/ceph/mgr/ceph-ceph-mgr1/keyring
[ceph-mgr1][INFO   ] Running command: sudo systemctl enable ceph-mgr@ceph-mgr1
[ceph-mgr1][WARNIN]                               Created           symlink
/etc/systemd/system/ceph-mgr.target.wants/ceph-mgr@ceph-mgr1.service →
/lib/systemd/system/ceph-mgr@.service.
[ceph-mgr1][INFO   ] Running command: sudo systemctl start ceph-mgr@ceph-mgr1
[ceph-mgr1][INFO   ] Running command: sudo systemctl enable ceph.target
```

3.4.13:验证 ceph-mgr 节点:

```
root@ceph-mgr1:~# ps -ef | grep ceph
root      27026      1  0 19:48 ?          00:00:00 /usr/bin/python3.6
/usr/bin/ceph-crash
ceph      28737      1 13 19:48 ?          00:00:07 /usr/bin/ceph-mgr -f --cluster
ceph --id ceph-mgr1 --setuser ceph --setgroup ceph
root     28922  1965  0 19:49 pts/0    00:00:00 grep --color=auto ceph
```

3.4.14:ceph-deploy 管理 ceph 集群:

在 ceph-deploy 节点配置一下系统环境，以方便后期可以执行 ceph 管理命令。

```
root@ceph-deploy:~# apt install ceph-common
root@ceph-deploy:~# yum install ceph-common
```

```
[ceph@ceph-deploy ceph-cluster]$ ceph-deploy admin ceph-deploy #推送正证书
给自己
```

```
[ceph_deploy.conf][DEBUG ] found configuration file at: /home/ceph/.cephdeploy.conf
[ceph_deploy.cli][INFO  ] Invoked (2.0.1): /bin/ceph-deploy admin ceph-deploy
[ceph_deploy.cli][INFO  ] ceph-deploy options:
[ceph_deploy.cli][INFO  ]   username           : None
[ceph_deploy.cli][INFO  ]   verbose            : False
[ceph_deploy.cli][INFO  ]   overwrite_conf     : False
[ceph_deploy.cli][INFO  ]   quiet              : False
[ceph_deploy.cli][INFO  ]   cd_conf            :
<ceph_deploy.conf.cephdeploy.Conf instance at 0x7faa926d6680>
```

```
[ceph_deploy.cli][INFO ] cluster : ceph
[ceph_deploy.cli][INFO ] client : ['ceph-deploy']
[ceph_deploy.cli][INFO ] func : <function admin at
0x7faa931ff230>
[ceph_deploy.cli][INFO ] ceph_conf : None
[ceph_deploy.cli][INFO ] default_release : False
[ceph_deploy.admin][DEBUG ] Pushing admin keys and conf to ceph-deploy
[ceph-deploy][DEBUG ] connection detected need for sudo
[ceph-deploy][DEBUG ] connected to host: ceph-deploy
[ceph-deploy][DEBUG ] detect platform information from remote host
[ceph-deploy][DEBUG ] detect machine type
[ceph-deploy][DEBUG ] write cluster configuration to /etc/ceph/{cluster}.conf
```

3.4.15: 测试 ceph 命令:

```
[root@ceph-deploy ~]# setfacl -m u:ceph:rw /etc/ceph/ceph.client.admin.keyring # 授权

ceph@ceph-deploy:/home/ceph/ceph-cluster$ ceph -s
cluster:
  id:      1883278f-95fe-4f85-b027-3a6eba444e61
  health: HEALTH_WARN
    mon is allowing insecure global_id reclaim #需要禁用非安全模式通信
    OSD count 0 < osd_pool_default_size 3 #集群的 OSD 数量小于 3

services:
  mon: 1 daemons, quorum ceph-mon1 (age 26m)
  mgr: ceph-mgr1(active, since 3m)
  osd: 0 osds: 0 up, 0 in

data:
  pools: 0 pools, 0 pgs
  objects: 0 objects, 0 B
  usage: 0 B used, 0 B / 0 B avail
  pgs:

ceph@ceph-deploy:/home/ceph/ceph-cluster$ ceph config set mon auth_allow_insecure_global_id_reclaim false
```

```
ceph@ceph-deploy:/home/ceph/ceph-cluster$ ceph versions
{
    "mon": {
        "ceph version 16.2.5 (0883bdea7337b95e4b611c768c0279868462204a) pacific (stable)": 1
    },
    "mgr": {
        "ceph version 16.2.5 (0883bdea7337b95e4b611c768c0279868462204a) pacific (stable)": 1
    },
    "osd": {},
    "mds": {},
    "overall": {
        "ceph version 16.2.5 (0883bdea7337b95e4b611c768c0279868462204a) pacific (stable)": 2
    }
}
ceph@ceph-deploy:/home/ceph/ceph-cluster$
```

杰哥的截图水印

```
[ceph@ceph-deploy ceph-cluster]$ ceph -s
cluster:
  id:      d7dc1d7a-ac99-4dc0-b551-e92149df7fde
  health: HEALTH_WARN
    OSD count 0 < osd_pool_default_size 3
```

```
services:
  mon: 1 daemons, quorum ceph-mon1
  mgr: ceph-mgr1(active)
  osd: 0 osds: 0 up, 0 in
```

```
data:
  pools: 0 pools, 0 pgs
  objects: 0 objects, 0 B
  usage: 0 B used, 0 B / 0 B avail
  pgs:
```

```
[ceph@ceph-deploy ceph-cluster]$
```

3.4.16:准备 OSD 节点:

OSD 节点安装运行环境:

```
[ceph@ceph-deploy ceph-cluster]$ ceph-deploy install --release pacific
ceph-node1 #擦除磁盘之前通过 deploy 节点对 node 节点执行安装 ceph 基本运行环境
[ceph@ceph-deploy ceph-cluster]$ ceph-deploy install --release pacific
ceph-node2
[ceph@ceph-deploy ceph-cluster]$ ceph-deploy install --release pacific
ceph-node3
[ceph@ceph-deploy ceph-cluster]$ ceph-deploy install --release pacific
ceph-node4
root@ceph-node2:~# apt autoremove #node 节点可选删除不需要的包
```

```
[ceph-node1] [DEBUG ] Setting up python3-pastedeploy (1.5.2-4) ...
[ceph-node1] [DEBUG ] Setting up python3-webtest (2.0.28-1ubuntu1) ...
[ceph-node1] [DEBUG ] Setting up python3-pecan (1.2.1-2) ...
[ceph-node1] [DEBUG ] update-alternatives: using /usr/bin/python3-pecan to provide /usr/bin/pecan (pecan) in auto mode
[ceph-node1] [DEBUG ] update-alternatives: using /usr/bin/python3-gunicorn_pecan to provide /usr/bin/gunicorn_pecan (gunicorn_pecan) in auto mode
[ceph-node1] [DEBUG ] Setting up ceph-mgr (16.2.5-1bionic) ...
[ceph-node1] [DEBUG ] Created symlink /etc/systemd/system/multi-user.target.wants/ceph-mgr.target -> /lib/systemd/system/ceph-mgr.target.
[ceph-node1] [DEBUG ] Created symlink /etc/systemd/system/ceph.target.wants/ceph-mgr.target -> /lib/systemd/system/ceph-mgr.target.
[ceph-node1] [DEBUG ] Setting up ceph (16.2.5-1bionic) ...
[ceph-node1] [DEBUG ] Processing triggers for systemd (237-3ubuntu10.43) ...
[ceph-node1] [DEBUG ] Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
[ceph-node1] [DEBUG ] Processing triggers for ureadahead (0.100.0-21) ...
[ceph-node1] [DEBUG ] Processing triggers for libc-bin (2.27-3ubuntu1.4) ...
[ceph-node1] [INFO ] Running command: sudo ceph --version
[ceph-node1] [DEBUG ] ceph version 16.2.5 (0883bdea7337b95e4b611c768c0279868462204a) pacific (stable)
ceph@ceph-deploy:/home/ceph/ceph-cluster$
```

列出 ceph node 节点磁盘:

```
[ceph@ceph-deploy ceph-cluster]$ ceph-deploy disk list ceph-node1 #列出远端存储 node 节点的磁盘信息
```

```
SecureCRT - 172.31.6.100-deploy - SecureCRT
[ceph@ceph-deploy:/home/ceph/ceph-cluster$ ceph-deploy disk list ceph-node1
[ceph_deploy.conf] [DEBUG ] found configuration file at: /var/lib/ceph/cephdeploy.conf
[ceph_deploy.cli] [INFO ] Invoked (2.0.1): /usr/bin/ceph-deploy disk list ceph-node1
[ceph_deploy.cli] [INFO ] ceph-deploy options:
[ceph_deploy.cli] [INFO ]   username          : None
[ceph_deploy.cli] [INFO ]   verbose           : False
[ceph_deploy.cli] [INFO ]   debug             : False
[ceph_deploy.cli] [INFO ]   overwrite_conf    : False
[ceph_deploy.cli] [INFO ]   subcommand        : list
[ceph_deploy.cli] [INFO ]   quiet             : False
[ceph_deploy.cli] [INFO ]   cd_conf            : <ceph_deploy.conf.cephdeploy.Conf instance at 0x7f5edef67f50>
[ceph_deploy.cli] [INFO ]   cluster            : ceph
[ceph_deploy.cli] [INFO ]   host               : ['ceph-node1']
[ceph_deploy.cli] [INFO ]   func               : <function disk at 0x7f5edef432d0>
[ceph_deploy.cli] [INFO ]   ceph_conf          : None
[ceph_deploy.cli] [INFO ]   default_release   : False
[ceph-node1] [DEBUG ] connection detected need for sudo
[ceph-node1] [DEBUG ] connected to host: ceph-node1
[ceph-node1] [DEBUG ] detect platform information from remote host
[ceph-node1] [DEBUG ] detect machine type
[ceph-node1] [DEBUG ] find the location of an executable
[ceph-node1] [INFO ] Running command: sudo fdisk -l
[ceph-node1] [INFO ] Disk /dev/sda: 120 GiB, 128849018880 bytes, 251658240 sectors
[ceph-node1] [INFO ] Disk /dev/sdb: 100 GiB, 107374182400 bytes, 209715200 sectors
[ceph-node1] [INFO ] Disk /dev/sdd: 100 GiB, 107374182400 bytes, 209715200 sectors
[ceph-node1] [INFO ] Disk /dev/sdf: 100 GiB, 107374182400 bytes, 209715200 sectors
[ceph-node1] [INFO ] Disk /dev/sdc: 100 GiB, 107374182400 bytes, 209715200 sectors
[ceph-node1] [INFO ] Disk /dev/sde: 100 GiB, 107374182400 bytes, 209715200 sectors
[ceph-node1] [INFO ] Disk /dev/sde: 100 GiB, 107374182400 bytes, 209715200 sectors
ceph@ceph-deploy:/home/ceph/ceph-cluster$
```

杰哥的截图水印

使用 ceph-deploy disk zap 擦除各 ceph node 的 ceph 数据磁盘:

ceph-node1 ceph-node2 ceph-node3 的存储节点磁盘擦除过程如下:

```
ceph@ceph-deploy:/home/ceph/ceph-cluster$ ceph-deploy disk zap ceph-node1
/dev/sdb
```

ceph-deploy disk zap ceph-node1 /dev/sdc

ceph-deploy disk zap ceph-node1 /dev/sdd

ceph-deploy disk zap ceph-node1 /dev/sde

ceph-deploy disk zap ceph-node1 /dev/sdf

ceph-deploy disk zap ceph-node2 /dev/sdb

ceph-deploy disk zap ceph-node2 /dev/sdc

ceph-deploy disk zap ceph-node2 /dev/sdd

ceph-deploy disk zap ceph-node2 /dev/sde

ceph-deploy disk zap ceph-node2 /dev/sdf

ceph-deploy disk zap ceph-node3 /dev/sdb

ceph-deploy disk zap ceph-node3 /dev/sdc

```
ceph-deploy disk zap ceph-node3 /dev/sdd
ceph-deploy disk zap ceph-node3 /dev/sde
ceph-deploy disk zap ceph-node3 /dev/sdf
```

```
ceph-deploy disk zap ceph-node4 /dev/sdb
ceph-deploy disk zap ceph-node4 /dev/sdc
ceph-deploy disk zap ceph-node4 /dev/sdd
ceph-deploy disk zap ceph-node4 /dev/sde
ceph-deploy disk zap ceph-node4 /dev/sdf
```

```
[ceph_deploy.conf] [DEBUG ] found configuration file at: /var/lib/ceph/.cephdeploy.conf
[ceph_deploy.cli] [INFO  ] Invoked (2.0.1): /usr/bin/ceph-deploy disk zap ceph-node4 /dev/sdf
[ceph_deploy.cli] [INFO  ] ceph-deploy options:
[ceph_deploy.cli] [INFO  ]   username           : None
[ceph_deploy.cli] [INFO  ]   verbose            : False
[ceph_deploy.cli] [INFO  ]   debug              : False
[ceph_deploy.cli] [INFO  ]   overwrite_conf     : False
[ceph_deploy.cli] [INFO  ]   subcommand         : zap
[ceph_deploy.cli] [INFO  ]   quiet              : False
[ceph_deploy.cli] [INFO  ]   cd_conf             : <ceph_deploy.conf.cephdeploy.Conf instance at 0x7f21fda93f50>
[ceph_deploy.cli] [INFO  ]   cluster             : ceph
[ceph_deploy.cli] [INFO  ]   host                : ceph-node4
[ceph_deploy.cli] [INFO  ]   func                : <function disk at 0x7f21fda6f2d0>
[ceph_deploy.cli] [INFO  ]   ceph_conf           : None
[ceph_deploy.cli] [INFO  ]   default_release    : False
[ceph_deploy.cli] [INFO  ]   disk                : ['/dev/sdf']
[ceph_deploy.osd] [DEBUG  ] zapping /dev/sdf on ceph-node4
[ceph-node4] [DEBUG  ] connection detected need for sudo
[ceph-node4] [DEBUG  ] connected to host: ceph-node4
[ceph-node4] [DEBUG  ] detect platform information from remote host
[ceph-node4] [DEBUG  ] detect machine type
[ceph-node4] [DEBUG  ] find the location of an executable
[ceph_deploy.osd] [INFO  ] Distro info: Ubuntu 18.04 bionic
[ceph-node4] [DEBUG  ] zeroing last few blocks of device
[ceph-node4] [DEBUG  ] find the location of an executable
[ceph-node4] [INFO  ] Running command: sudo /usr/sbin/ceph-volume lvm zap /dev/sdf
[ceph-node4] [WARNIN] --> Zapping: /dev/sdf
[ceph-node4] [WARNIN] --> --destroy was not specified, but zapping a whole device will remove the partition table
[ceph-node4] [WARNIN] Running command: /bin/dd if=/dev/zero of=/dev/sdf bs=1M count=10 conv=fsync
[ceph-node4] [WARNIN] stderr: 10+0 records in
[ceph-node4] [WARNIN] 10+0 records out
[ceph-node4] [WARNIN] 10485760 bytes (10 MB, 10 MiB) copied, 0.0263012 s, 399 MB/s
[ceph-node4] [WARNIN] --> Zapping successful for: <Raw Device: /dev/sdf>
ceph@ceph-deploy:/home/ceph/ceph-clusters$
```

马哥教育,杰哥的截图水印

3.4.17:添加 OSD:

数据分类保存方式:

Data: 即 ceph 保存的对象数据

Block: rocks DB 数据即元数据

block-wal: 数据库的 wal 日志

- 单块磁盘:
 - 机械硬盘或者SSD:
 - data: 即ceph保存的对象数据
 - block: rocks DB数据即元数据
 - block-wal: 数据库的wal日志
- 两块磁盘:
 - SSD:
 - block: rocks DB数据即元数据
 - block-wal: 数据库的wal日志
 - 机械硬盘:
 - data: 即ceph保存的对象数据
- 三块磁盘:
 - NVME:
 - block: rocks DB数据即元数据
 - SSD:
 - block-wal: 数据库的wal日志
 - 机械硬盘:
 - data: 即ceph保存的对象数据

```
# ceph-deploy osd create {node} --data /path/to/data --block-db /path/to/db-device
# ceph-deploy osd create {node} --data /path/to/data --block-wal /path/to/wal-device
# ceph-deploy osd create {node} --data /path/to/data --block-db /path/to/db-device --block-wal
/path/to/wal-device
```

```
[ceph@ceph-deploy ceph-cluster]$ ceph-deploy osd --help
usage: ceph-deploy osd [-h] {list,create} ...
```

Create OSDs from a data disk on a remote host:

```
ceph-deploy osd create {node} --data /path/to/device
```

For bluestore, optional devices can be used::

```
ceph-deploy osd create {node} --data /path/to/data --block-db /path/to/db-device
ceph-deploy osd create {node} --data /path/to/data --block-wal /path/to/wal-device
ceph-deploy osd create {node} --data /path/to/data --block-db /path/to/db-device
--block-wal /path/to/wal-device
```

For filestore, the journal must be specified, as well as the objectstore::

```
ceph-deploy osd create {node} --filestore --data /path/to/data --journal
/path/to/journal #使用 filestor 的数据和文件系统的日志的路径
```

For data devices, it can be an existing logical volume in the format of:
vg/lv, or a device. For other OSD components like wal, db, and journal, it
can be logical volume (in vg/lv format) or it must be a GPT partition.

positional arguments:

{list,create}

list List OSD info from remote host(s)

create Create new Ceph OSD daemon by preparing and activating a

```
device
```

optional arguments:

```
-h, --help      show this help message and exit
```

添加 OSD:

```
[ceph@ceph-deploy ceph-cluster]$ ceph-deploy osd create ceph-node1 --data /dev/sdb
```

```
[ceph@ceph-deploy ceph-cluster]$ ceph-deploy osd create ceph-node1 --data /dev/sdc
```

```
[ceph@ceph-deploy ceph-cluster]$ ceph-deploy osd create ceph-node1 --data /dev/sdd
```

```
[ceph@ceph-deploy ceph-cluster]$ ceph-deploy osd create ceph-node1 --data /dev/sde
```

```
[ceph@ceph-deploy ceph-cluster]$ ceph-deploy osd create ceph-node1 --data /dev/sdf
```

```
[ceph@ceph-deploy ceph-cluster]$ ceph-deploy osd create ceph-node2 --data /dev/sdb
```

```
[ceph@ceph-deploy ceph-cluster]$ ceph-deploy osd create ceph-node2 --data /dev/sdc
```

```
[ceph@ceph-deploy ceph-cluster]$ ceph-deploy osd create ceph-node2 --data /dev/sdd
```

```
[ceph@ceph-deploy ceph-cluster]$ ceph-deploy osd create ceph-node2 --data /dev/sde
```

```
[ceph@ceph-deploy ceph-cluster]$ ceph-deploy osd create ceph-node2 --data /dev/sdf
```

```
[ceph@ceph-deploy ceph-cluster]$ ceph-deploy osd create ceph-node3 --data /dev/sdb
```

```
[ceph@ceph-deploy ceph-cluster]$ ceph-deploy osd create ceph-node3 --data /dev/sdc
```

```
[ceph@ceph-deploy ceph-cluster]$ ceph-deploy osd create ceph-node3 --data /dev/sdd
```

```
[ceph@ceph-deploy ceph-cluster]$ ceph-deploy osd create ceph-node3 --data /dev/sde
```

```
[ceph@ceph-deploy ceph-cluster]$ ceph-deploy osd create ceph-node3 --data /dev/sdf
```

```
..... 其他主机的磁盘以此类推添加!
```

```
[ceph-node1] [WARNING] Running command: /bin/ceph --cluster ceph --name client.bootstrap-osd --keyring /var/lib/ceph/bootstrap-osd/ceph.keyring mon getmap -o /var/lib/ceph/osd/ceph.0/activate.monmap
[ceph-node1] [WARNING] stderr: get monmap epoch 1
[ceph-node1] [WARNING] Running command: /bin/ceph authtool /var/lib/ceph/osd/ceph-0/keyring --create-keyring --name osd.0 --add-key AQAFzLvgp5C0FhAAyNQ6ezS2xpRAYWArxCeHew==
[ceph-node1] [WARNING] stdout: creating /var/lib/ceph/osd/ceph-0/keyring
[ceph-node1] [WARNING] addentity entity.osd.0 auth.(auid = 18446744073709551615 key=AQAFzLvgp5C0FhAAyNQ6ezS2xpRAYWArxCeHew== with 0 caps)
[ceph-node1] [WARNING] Running command: /bin/chown -R ceph:ceph /var/lib/ceph/osd/ceph-0/keyring
[ceph-node1] [WARNING] Running command: /bin/chown -R ceph:ceph /var/lib/ceph/osd/ceph-0/
[ceph-node1] [WARNING] Running command: /bin/ceph OSD --cluster ceph --osd-objectstore bluestore --mkfs -i 0 -m monmap /var/lib/ceph/osd/ceph-0/activate.monmap --keyfile - --osd-data /var/lib/ceph/osd/ceph-0/ --osd-uuid 038e74bc-5deb-41ba-948c-18f5c50d450c --setuser ceph --setgroup ceph
[ceph-node1] [WARNING] --> ceph-volume lvm prepare successful for: /dev/sdb
[ceph-node1] [WARNING] Running command: /bin/chown -R ceph:ceph /var/lib/ceph/osd/ceph-0
[ceph-node1] [WARNING] [WARNING] Running command: /bin/ceph-bluestore-tool --cluster=ceph prime-osd-dir --dev /dev/ceph-d00ced54-4151-438b-ad84-1eb80e990e79/osd-block-038e74bc-5deb-41ba-a-948c-18f5c50d450c --path /var/lib/ceph/osd/ceph-0 --no-mon-config
[ceph-node1] [WARNING] Running command: /bin/ln -snf /dev/ceph-d00ced54-4151-438b-ad84-1eb80e990e79/osd-block-038e74bc-5deb-41ba-948c-18f5c50d450c /var/lib/ceph/osd/ceph-0/block
[ceph-node1] [WARNING] Running command: /bin/chown -h ceph:ceph /var/lib/ceph/osd/ceph-0/block
[ceph-node1] [WARNING] Running command: /bin/chown -R ceph:ceph /dev/dm-2
[ceph-node1] [WARNING] Running command: /bin/chown -R ceph:ceph /var/lib/ceph/osd/ceph-0
[ceph-node1] [WARNING] Running command: /bin/systemctl enable ceph-volume@lvm-0-038e74bc-5deb-41ba-948c-18f5c50d450c
[ceph-node1] [WARNING] stderr: Created symlink from /etc/systemd/system/multi-user.target.wants/ceph-volume@lvm-0-038e74bc-5deb-41ba-948c-18f5c50d450c.service to /usr/lib/systemd/system/ceph-volume@.service.
[ceph-node1] [WARNING] Running command: /bin/systemctl enable --runtime ceph-osd@0
[ceph-node1] [WARNING] stderr: Created symlink from /run/systemd/system/ceph-osd.target.wants/ceph-osd@0.service to /usr/lib/systemd/system/ceph-osd@.service.
[ceph-node1] [WARNING] Running command: /bin/systemctl start ceph-osd@0
[ceph-node1] [WARNING] [WARNING] --> ceph-volume lvm activate successful for osd ID: 0
[ceph-node1] [WARNING] --> ceph-volume lvm create successful for: /dev/sdb
[ceph-node1] [INFO] [INFO] checking OSD status...
[ceph-node1] [DEBUG] [DEBUG] find the location of an executable
[ceph-node1] [INFO] [INFO] Running command: sudo /bin/ceph --cluster=ceph osd stat --format=json
[ceph_deploy.osd] [DEBUG] Host ceph-node1 is now ready for osd use.
[ceph@ceph-deploy ceph-cluster]$
```

3.4.18:设置 OSD 服务自启动:

默认就已经为自启动, node 节点添加完成后, 开源测试 node 服务器重启后, OSD 是否会自动启动。

```
ceph@ceph-deploy:/home/ceph/ceph-cluster$ ceph -s
cluster:
  id: 67ecc7e9-c7c8-47d8-ba1a-4db1206a953c
  health: HEALTH_WARN
    5 osds down
    1 host (5 osds) down
  services:
    mon: 1 daemons, quorum ceph-mon1 (age 40m)
    mgr: ceph-mgr1(active, since 34m)
    osd: 20 osds: 15 up (since 16s), 20 in (since 80s)

  data:
    pools: 1 pools, 1 pgs
    objects: 0 objects, 0 B
    usage: 152 MiB used, 2.0 TiB / 2.0 TiB avail
    pgs: 1 active+undersized

ceph@ceph-deploy:/home/ceph/ceph-cluster$
```

```
[root@ceph-node1 ~]# ps -ef | grep osd
```

```
[root@ceph-node1 ~]# systemctl enable ceph-osd@0 ceph-osd@1 ceph-osd@2
ceph-osd@3 ceph-osd@4
Created symlink from /etc/systemd/system/ceph-osd.target.wants/ceph-osd@0.service
to /usr/lib/systemd/system/ceph-osd@.service.
Created symlink from /etc/systemd/system/ceph-osd.target.wants/ceph-osd@1.service
to /usr/lib/systemd/system/ceph-osd@.service.
Created symlink from /etc/systemd/system/ceph-osd.target.wants/ceph-osd@2.service
to /usr/lib/systemd/system/ceph-osd@.service.
Created symlink from /etc/systemd/system/ceph-osd.target.wants/ceph-osd@3.service
to /usr/lib/systemd/system/ceph-osd@.service.
```

```
Created symlink from /etc/systemd/system/ceph-osd.target.wants/ceph-osd@4.service
to /usr/lib/systemd/system/ceph-osd@.service.
```

```
[root@ceph-node2 ~]# ps -ef | grep osd
[root@ceph-node2 ~]# systemctl enable ceph-osd@5 ceph-osd@6 ceph-osd@7
ceph-osd@8 ceph-osd@9
Created symlink from /etc/systemd/system/ceph-osd.target.wants/ceph-osd@5.service
to /usr/lib/systemd/system/ceph-osd@.service.
Created symlink from /etc/systemd/system/ceph-osd.target.wants/ceph-osd@6.service
to /usr/lib/systemd/system/ceph-osd@.service.
Created symlink from /etc/systemd/system/ceph-osd.target.wants/ceph-osd@7.service
to /usr/lib/systemd/system/ceph-osd@.service.
Created symlink from /etc/systemd/system/ceph-osd.target.wants/ceph-osd@8.service
to /usr/lib/systemd/system/ceph-osd@.service.
Created symlink from /etc/systemd/system/ceph-osd.target.wants/ceph-osd@9.service
to /usr/lib/systemd/system/ceph-osd@.service.
```

```
[root@ceph-node3 ~]# ps -ef | grep osd
[root@ceph-node3 ~]# systemctl enable ceph-osd@10 ceph-osd@11 ceph-osd@12
ceph-osd@13 ceph-osd@14
Created           symlink      from
/etc/systemd/system/ceph-osd.target.wants/ceph-osd@10.service          to
/usr/lib/systemd/system/ceph-osd@.service.
Created           symlink      from
/etc/systemd/system/ceph-osd.target.wants/ceph-osd@11.service          to
/usr/lib/systemd/system/ceph-osd@.service.
Created           symlink      from
/etc/systemd/system/ceph-osd.target.wants/ceph-osd@12.service          to
/usr/lib/systemd/system/ceph-osd@.service.
Created           symlink      from
/etc/systemd/system/ceph-osd.target.wants/ceph-osd@13.service          to
/usr/lib/systemd/system/ceph-osd@.service.
Created           symlink      from
/etc/systemd/system/ceph-osd.target.wants/ceph-osd@14.service          to
/usr/lib/systemd/system/ceph-osd@.service.
```

```
[root@ceph-node3 ~]# systemctl enable ceph-osd@10 ceph-osd@11 ceph-osd@12 ceph-osd@13 ceph-osd@14
Created symlink from /etc/systemd/system/ceph-osd.target.wants/ceph-osd@10.service to /usr/lib/systemd/system/ceph-osd@.service.
Created symlink from /etc/systemd/system/ceph-osd.target.wants/ceph-osd@11.service to /usr/lib/systemd/system/ceph-osd@.service.
Created symlink from /etc/systemd/system/ceph-osd.target.wants/ceph-osd@12.service to /usr/lib/systemd/system/ceph-osd@.service.
Created symlink from /etc/systemd/system/ceph-osd.target.wants/ceph-osd@13.service to /usr/lib/systemd/system/ceph-osd@.service.
Created symlink from /etc/systemd/system/ceph-osd.target.wants/ceph-osd@14.service to /usr/lib/systemd/system/ceph-osd@.service.
[root@ceph-node3 ~]#
[root@ceph-node3 ~]# systemctl status ceph-osd@10 ceph-osd@11 ceph-osd@12 ceph-osd@13 ceph-osd@14
● ceph-osd@10.service - Ceph object storage daemon osd.10
  Loaded: loaded (/usr/lib/systemd/system/ceph-osd@.service; enabled; vendor preset: disabled)
    Active: active (running) since Tue 2021-06-01 14:35:54 CST; 1min ago
      Main PID: 3584 (ceph-osd)
        CGroup: /system.slice/system-ceph\x2dosd.slice/ceph-osd@10.service
                  └─3584 /usr/bin/ceph-osd -f --cluster ceph --id 10 --setuser ceph --setgroup ceph

Jun 01 14:35:54 ceph-node3.jie.local systemd[1]: Starting Ceph object storage daemon osd.10...
Jun 01 14:35:54 ceph-node3.jie.local systemd[1]: Started Ceph object storage daemon osd.10.
Jun 01 14:35:54 ceph-node3.jie.local ceph-osd[3584]: starting osd.10 at - osd_data /var/lib/ceph/osd/ceph-10 /var/lib/ceph/ceph-10/journal
Jun 01 14:35:56 ceph-node3.jie.local ceph-osd[3584]: 2021-06-01 14:35:54.675 7f2184220b80 -1 osd.10 0 log_to_monitors {default=true}
Jun 01 14:35:56 ceph-node3.jie.local ceph-osd[3584]: 2021-06-01 14:35:56.670 7f215d7700 -1 osd.10 0 waiting for initial osdmap

● ceph-osd@11.service - Ceph object storage daemon osd.11
  Loaded: loaded (/usr/lib/systemd/system/ceph-osd@.service; enabled; vendor preset: disabled)
    Active: active (running) since Tue 2021-06-01 14:36:33 CST; 10min ago
      Main PID: 4086 (ceph-osd)
        CGroup: /system.slice/system-ceph\x2dosd.slice/ceph-osd@11.service
                  └─4086 /usr/bin/ceph-osd -f --cluster ceph --id 11 --setuser ceph --setgroup ceph

Jun 01 14:36:33 ceph-node3.jie.local systemd[1]: Starting Ceph object storage daemon osd.11...
Jun 01 14:36:33 ceph-node3.jie.local systemd[1]: Started Ceph object storage daemon osd.11.
Jun 01 14:36:33 ceph-node3.jie.local ceph-osd[4086]: starting osd.11 at - osd_data /var/lib/ceph/osd/ceph-11 /var/lib/ceph/ceph-11/journal
Jun 01 14:36:33 ceph-node3.jie.local ceph-osd[4086]: 2021-06-01 14:36:33.895 7f7f314e3b80 -1 osd.11 0 log_to_monitors {default=true}
Jun 01 14:36:35 ceph-node3.jie.local ceph-osd[4086]: 2021-06-01 14:36:35.634 7f7f0aa9a700 -1 osd.11 0 waiting for initial osdmap

● ceph-osd@12.service - Ceph object storage daemon osd.12
  Loaded: loaded (/usr/lib/systemd/system/ceph-osd@.service; enabled; vendor preset: disabled)
```

3.4.20:验证 ceph 集群:

```
[ceph@ceph-deploy ceph-cluster]$ ceph health
HEALTH_OK
[ceph@ceph-deploy ceph-cluster]$ [ceph@ceph-deploy ceph-cluster]$ ceph -s
cluster:
  id: d7dc1d7a-ac99-4dc0-b551-e92149df7fde
  health: HEALTH_OK

  services:
    mon: 1 daemons, quorum ceph-mon1
    mgr: ceph-mgr1(active)
    osd: 15 osds: 15 up, 15 in

  data:
    pools: 0 pools, 0 pgs
    objects: 0 objects, 0 B
    usage: 15 GiB used, 1.5 TiB / 1.5 TiB avail
    pgs:

[ceph@ceph-deploy ceph-cluster]$
```

杰哥的截图水印

3.4.20:从 RADOS 移除 OSD:

Ceph 集群中的一个 OSD 是一个 node 节点的服务进程且对应于一个物理磁盘设备, 是一个专用的守护进程。在某 OSD 设备出现故障, 或管理员出于管理之需确实要移除特定的 OSD 设备时, 需要先停止相关的守护进程, 而后再进行移除操作。对于 Luminous 及其之后的版本来说, 停止和移除命令的格式分别如下所示:

1. 停用设备: `ceph osd out {osd-num}`
2. 停止进程: `sudo systemctl stop ceph-osd@{osd-num}`
3. 移除设备: `ceph osd purge {id} --yes-i-really-mean-it`

若类似如下的 OSD 的配置信息存在于 `ceph.conf` 配置文件中, 管理员在删除 OSD 之后手动将其删除。

不过，对于 Luminous 之前的版本来说，管理员需要依次手动执行如下步骤删除 OSD 设备：

1. 于 CRUSH 运行图中移除设备： ceph osd crush remove {name}
2. 移除 OSD 的认证 key： ceph auth del osd.{osd-num}
3. 最后移除 OSD 设备： ceph osd rm {osd-num}

3.4.21: 测试上传与下载数据：

存取数据时，客户端必须首先连接至 RADOS 集群上某存储池，然后根据对象名称由相关的 CRUSH 规则完成数据对象寻址。于是，为了测试集群的数据存取功能，这里首先创建一个用于测试的存储池 mypool，并设定其 PG 数量为 32 个。

```
$ ceph -h #一个更底层的命令  
$ rados -h #
```

创建 pool:

```
[ceph@ceph-deploy ceph-cluster]$ ceph osd pool create mypool 32 32 #32PG 和  
32PGP  
pool 'mypool' created  
[ceph@ceph-deploy ceph-cluster]$ ceph pg ls-by-pool mypool | awk '{print  
$1,$2,$15}' #验证 PG 与 PGP 组合  
  
[ceph@ceph-deploy ceph-cluster]$ ceph osd pool ls  
mypool  
或者：  
[ceph@ceph-deploy ceph-cluster]$ rados lspools  
mypool
```

当前的 ceph 环境还没部署使用块存储和文件系统使用 ceph，也没有使用对象存储的客户端，但是 ceph 的 rados 命令可以实现访问 ceph 对象存储的功能：

上传文件：

```
[ceph@ceph-deploy ceph-cluster]$ sudo rados put msg1 /var/log/syslog  
--pool=mypool #把 messages 文件上传到 mypool 并指定对象 id 为 msg1
```

列出文件：

```
[ceph@ceph-deploy ceph-cluster]$ rados ls --pool=mypool  
msg1
```

文件信息：

ceph osd map 命令可以获取到存储池中数据对象的具体位置信息：

```
ceph@ceph-deploy:/home/ceph/ceph-cluster$ ceph osd map mypool msg1  
osdmap e114 pool 'mypool' (2) object 'msg1' -> pg 2.c833d430 (2.10) -> up ([15,13,0],  
p15) acting ([15,13,0], p15)
```

表示文件放在了存储池 id 为 2 的 c833d430 的 PG 上，10 为当前 PG 的 id，2.10 表示数据是在 id 为 2 的存储池当中 id 为 10 的 PG 中存储，在线的 OSD 编号 15,13,10，主 OSD 为 5，活动的 OSD 15,13,10，三个 OSD 表示数据放一共 3 个副本，PG 中的 OSD 是 ceph 的 crush 算法计算出三份数据保存在哪些 OSD。

下载文件:

```
[ceph@ceph-deploy ceph-cluster]$ sudo rados get msg1 --pool=mypool /opt/my.txt
[ceph@ceph-deploy ceph-cluster]$ ll /opt/
total 180
-rw-r--r-- 1 root root 180766 Jun  1 15:01 my.txt
```

验证下载文件:

```
[ceph@ceph-deploy ceph-cluster]$ head /opt/my.txt
ceph@ceph-deploy:/home/ceph/ceph-cluster$ head /opt/my.txt
Sep 15 16:13:30 ubuntu systemd-modules-load[494]: Inserted module 'iscsi_tcp'
Sep 15 16:13:30 ubuntu systemd-modules-load[494]: Inserted module 'ib_iser'
Sep 15 16:13:30 ubuntu systemd-udevd[517]: Network interface NamePolicy= disabled on kernel command line, ignoring...
Sep 15 16:13:30 ubuntu systemd[1]: Starting Flush Journal to Persistent Storage...
Sep 15 16:13:30 ubuntu systemd[1]: Activated swap /swapfile.
Sep 15 16:13:30 ubuntu systemd[1]: Reached target Swap.
Sep 15 16:13:30 ubuntu systemd[1]: Started Set the console keyboard layout.
Sep 15 16:13:30 ubuntu systemd[1]: Reached target Local File Systems (Pre).
Sep 15 16:13:30 ubuntu systemd[1]: Reached target Local File Systems.
Sep 15 16:13:30 ubuntu systemd[1]: Starting AppArmor initialization...
ceph@ceph-deploy:/home/ceph/ceph-cluster$
```

修改文件:

```
[ceph@ceph-deploy ceph-cluster]$ sudo rados put msg1 /etc/passwd --pool=mypool
[ceph@ceph-deploy ceph-cluster]$ sudo rados get msg1 --pool=mypool /opt/2.txt
```

验证下载文件:

```
[ceph@ceph-deploy ceph-cluster]$ tail /opt/2.txt
ceph@ceph-deploy:/home/ceph/ceph-cluster$ tail /opt/2.txt
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
lxde:x:105:65534::/var/lib/lxde/:/bin/false
uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin
sshd:x:109:65534::/run/sshd:/usr/sbin/nologin
pollinate:x:110:1::/var/cache/pollinate:/bin/false
jack:x:1000:1000:jack,,,,:/home/jack:/bin/bash
ceph:x:2022:2022:Ceph storage service:/var/lib/ceph:/bin/bash
ceph@ceph-deploy:/home/ceph/ceph-cluster$
ceph@ceph-deploy:/home/ceph/ceph-cluster$
```

删除文件:

```
[ceph@ceph-deploy ceph-cluster]$ sudo rados rm msg1 --pool=mypool
[ceph@ceph-deploy ceph-cluster]$ rados ls --pool=mypool
```

3.3:扩展 ceph 集群实现高可用:

主要是扩展 ceph 集群的 mon 节点以及 mgr 节点，以实现集群高可用。

3.3.1:扩展 ceph-mon 节点:

Ceph-mon 是原生具备自选举以实现高可用机制的 ceph 服务，节点数量通常是奇数。

```
root@ceph-mon2:~# apt install ceph-mon
root@ceph-mon3:~# apt install ceph-mon
```

```
[root@ceph-mon2 ~]# yum install ceph-common ceph-mon
[ceph@ceph-deploy ceph-cluster]$ ceph-deploy mon add ceph-mon2
[ceph-mon2][DEBUG ]    "name": "ceph-mon2",
[ceph-mon2][DEBUG ]    "outside_quorum": [],
[ceph-mon2][DEBUG ]    "quorum": [],
[ceph-mon2][DEBUG ]    "rank": -1,
[ceph-mon2][DEBUG ]    "state": "probing",
[ceph-mon2][DEBUG ]    "sync_provider": []
[ceph-mon2][DEBUG ] }
[ceph-mon2][DEBUG ] ****
[ceph-mon2][INFO  ] monitor: mon.ceph-mon2 is currently at the state of probing
```

```
[root@ceph-mon3 ~]# yum install ceph-common ceph-mon
[ceph@ceph-deploy ceph-cluster]$ ceph-deploy mon add ceph-mon3
[ceph-mon3][DEBUG ]    "name": "ceph-mon3",
[ceph-mon3][DEBUG ]    "outside_quorum": [],
[ceph-mon3][DEBUG ]    "quorum": [],
[ceph-mon3][DEBUG ]    "rank": -1,
[ceph-mon3][DEBUG ]    "state": "probing",
[ceph-mon3][DEBUG ]    "sync_provider": []
[ceph-mon3][DEBUG ] }
[ceph-mon3][DEBUG ] ****
[ceph-mon3][INFO  ] monitor: mon.ceph-mon3 is currently at the state of probing
```

3.3.2:验证 ceph-mon 状态:

```
[ceph@ceph-deploy ceph-cluster]$ ceph quorum_status
[ceph@ceph-deploy ceph-cluster]$ ceph quorum_status --format json-pretty
```

```
[ceph@ceph-deploy ceph-cluster]$ ceph quorum_status --format json-pretty
{
    "election_epoch": 12,
    "quorum": [
        0,
        1,
        2
    ],
    "quorum_names": [
        "ceph-mon1",
        "ceph-mon2",
        "ceph-mon3"
    ],
    "quorum_leader_name": "ceph-mon1", 当前的leader
    "monmap": {
        "epoch": 3,
        "fsid": "d7dc1d7a-ac99-4dc0-b551-e92149df7fde",
        "modified": "2021-06-01 15:20:23.616409",
        "created": "2021-06-01 11:00:07.334530",
        "features": [
            "persistent": [
                "kraken",
                "luminous",
                "mimic",
                "osdmap-prune"
            ],
            "optional": []
        ],
        "mons": [
            {
                "rank": 0,
                "name": "ceph-mon1",
                "addr": "172.31.6.104:6789/0",
                "public_addr": "172.31.6.104:6789/0"
            },
            {
                "rank": 1, 当前节点的等级
                "name": "ceph-mon2",
                "addr": "172.31.6.105:6789/0", 节点名称
                "public_addr": "172.31.6.105:6789/0"
            },
            {
                "rank": 2,
                "name": "ceph-mon3",
                "addr": "172.31.6.106:6789/0",
                "public_addr": "172.31.6.106:6789/0" 监听地址
            }
        ]
    }
}
[ceph@ceph-deploy ceph-cluster]$
```

验证 ceph 集群状态:

```
[ceph@ceph-deploy ceph-cluster]$ ceph -s
cluster:
  id:      d7dc1d7a-ac99-4dc0-b551-e92149df7fde
  health:  HEALTH_WARN
          too few PGs per OSD (6 < min 30) 杰哥的截图水印

  services:
    mon: 3 daemons, quorum ceph-mon1,ceph-mon2,ceph-mon3
    mgr: ceph-mgr1(active)
    osd: 15 osds: 15 up, 15 in

  data:
    pools:   1 pools, 32 pgs
    objects: 0 objects, 0 B
    usage:   15 GiB used, 1.5 TiB / 1.5 TiB avail
    pgs:     32 active+clean

[ceph@ceph-deploy ceph-cluster]$
```

3.3.3: 扩展 mgr 节点:

安装:

```
~# apt install ceph-mgr
```

```
[root@ceph-mgr2 ~]# yum install ceph-mgr -y
```

添加:

```
[ceph@ceph-deploy ceph-cluster]$ ceph-deploy mgr create ceph-mgr2
[ceph@ceph-deploy ceph-cluster]$ ceph-deploy admin ceph-mgr2 #同步配置文件到
ceph-mg2 节点
```

3.3.4:验证 mgr 节点状态:

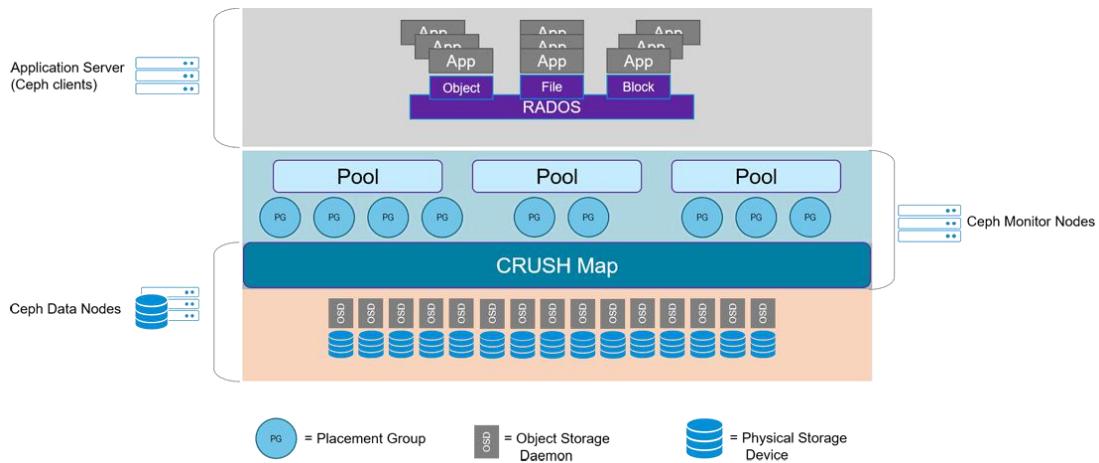
```
[ceph@ceph-deploy ceph-cluster]$ ceph -s
cluster:
  id:      d7dc1d7a-ac99-4dc0-b551-e92149df7fde
  health:  HEALTH_WARN
            too few PGs per OSD (6 < min 30) 杰哥的截图水印
  services:
    mon: 3 daemons, quorum ceph-mon1,ceph-mon2,ceph-mon3
    mgr: ceph-mgr1(active), standbys: ceph-mgr2
    osd: 15 osds: 15 up, 15 in

  data:
    pools:   1 pools, 32 pgs
    objects: 0 objects, 0 B
    usage:   15 GiB used, 1.5 TiB / 1.5 TiB avail
    pgs:     32 active+clean

[ceph@ceph-deploy ceph-cluster]$
```

四：ceph 集群应用基础：

ceph 的集群应用



4.1: 块设备 RBD:

RBD(RADOS Block Devices)即为块存储的一种, RBD 通过 librbd 库与 OSD 进行交互, RBD 为 KVM 等虚拟化技术和云服务 (如 OpenStack 和 CloudStack) 提供高性能和无限可扩展性的存储后端, 这些系统依赖于 libvirt 和 QEMU 实用程序与 RBD 进行集成, 客户端基于 librbd 库即可将 RADOS 存储集群用作块设备, 不过, 用于 rbd 的存储池需要事先启用 rbd 功能并进行初始化。例如, 下面的命令创建一个名为 myrbd1 的存储池, 并在启用 rbd 功能后对其进行初始化:

4.1.1: 创建 RBD:

具体 RBD 使用会在第六章详细介绍:

创建存储池命令格式:

```
$ ceph osd pool create <poolname> pg_num ppg_num {replicated|erasure}
```

```
$ ceph osd pool create myrbd1 64 64 # 创建存储池, 指定 pg 和 ppg 的数量, ppg 是对存在于 pg 的数据进行组合存储, ppg 通常等于 pg 的值  
pool 'myrbd1' created
```

```
$ ceph osd pool --help  
$ ceph osd pool application enable myrbd1 rbd # 对存储池启用 RBD 功能  
enabled application 'rbd' on pool 'myrbd1'
```

```
$ rbd -h  
$ rbd pool init -p myrbd1 # 通过 RBD 命令对存储池初始化
```

4.1.2:创建并验证 img:

不过，`rbd` 存储池并不能直接用于块设备，而是需要事先在其中按需创建映像（image），并把映像文件作为块设备使用，`rbd` 命令可用于创建、查看及删除块设备相在的映像（image），以及克隆映像、创建快照、将映像回滚到快照和查看快照等管理操作，例如，下面的命令能够创建一个名为 `myimg1` 的映像：

```
$ rbd create myimg1 --size 5G --pool myrbd1
$ rbd create myimg2 --size 3G --pool myrbd1 --image-format 2 --image-feature
layering
#后续步骤会使用 myimg2，由于 centos 系统内核较低无法挂载使用，因此只开启部分特
性。
除了 layering 其他特性需要高版本内核支持

$ rbd ls --pool myrbd1 #列出指定的 pool 中所有的 img
myimg1
myimg2

$ rbd --image myimg1 --pool myrbd1 info #查看指定 rdb 的信息
rbd image 'myimg1':
    size 5 GiB in 1280 objects
    order 22 (4 MiB objects)
    id: 116a6b8b4567
    block_name_prefix: rbd_data.116a6b8b4567
    format: 2
    features: layering, exclusive-lock, object-map, fast-diff, deep-flatten
    op_features:
    flags:
    create_timestamp: Thu Oct 22 11:25:14 2020

$ rbd --image myimg2 --pool myrbd1 info
rbd image 'myimg2':
    size 1 GiB in 256 objects
    order 22 (4 MiB objects)
    id: 11b26b8b4567
    block_name_prefix: rbd_data.11b26b8b4567
    format: 2
    features: layering
    op_features:
    flags:
    create_timestamp: Thu Oct 22 12:01:20 2020
```

4.1.3:客户端使用块存储:

4.1.3.1:当前 ceph 状态:

```
[ceph@ceph-deploy ceph-cluster]$ ceph df
--- RAW STORAGE --GLOBAL:
      SIZE      AVAIL      RAW USED      %RAW USED
      1.5 TiB    1.5 TiB     15 GiB       1.01
POOLS:
      NAME      ID      USED      %USED      MAX AVAIL      OBJECTS
  mypool      1        0 B        0        470 GiB         0
  myrdb1      2     405 B        0        470 GiB         7
```

4.1.3.2:在客户端安装 ceph-common:

客户端服务器配置 yum 源及 ceph 认证文件:

```
#配置 yum 源:
# yum install epel-release
# yum install
https://mirrors.aliyun.com/ceph/rpm-octopus/el7/noarch/ceph-release-1-1.el7.noarch.rpm -y

#yum install ceph-common

#从部署服务器同步认证文件:
[ceph@ceph-deploy ceph-cluster]$ scp ceph.conf ceph.client.admin.keyring
root@172.31.6.201:/etc/ceph/
```

4.1.3.3:客户端映射 img:

```
[root@ceph-client ceph]# rbd -p myrbd1 map myimg2
/dev/rbd0
[root@ceph-client ceph]# rbd -p myrdb1 map myimg1
rbd: sysfs write failed
RBD image feature set mismatch. You can disable features unsupported by the kernel
with "rbd feature disable myrdb1/myimg1 object-map fast-diff deep-flatten".
In some cases useful info is found in syslog - try "dmesg | tail".
rbd: map failed: (6) No such device or address
```

4.1.3.4:客户端验证 RBD:

```
[root@ceph-client ~]# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda          8:0    0 120G  0 disk
└─sda1      8:1    0   512M 0 part /boot
```

```
|└─sda2          8:2    0 119.5G  0 part
|   ├─centos-root 253:0    0 119.5G  0 lvm  /
|   └─centos-swap 253:1    0     4M  0 lvm  [SWAP]
sr0           11:0    1     1G  0 rom
rbd0          252:0    0     1G  0 disk
```

```
[root@ceph-client ~]# fdisk -l /dev/rbd0
```

```
Disk /dev/rbd0: 1073 MB, 1073741824 bytes, 2097152 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 4194304 bytes / 4194304 bytes
```

4.1.3.5:客户端格式化磁盘并挂载使用：

```
[root@ceph-client ~]# mkfs.xfs /dev/rbd0
meta-data=/dev/rbd0              isize=512    agcount=8, agsize=32768 blks
                                =                      sectsz=512  attr=2, projid32bit=1
                                =                      crc=1      finobt=0, sparse=0
data     =                      bsize=4096   blocks=262144, imaxpct=25
        =                      sunit=1024   swidth=1024 blks
naming   =version 2             bsize=4096   ascii-ci=0 ftype=1
log      =internal log          bsize=4096   blocks=2560, version=2
        =                      sectsz=512   sunit=8 blks, lazy-count=1
realtime =none                  extsz=4096   blocks=0, rtextents=0
[root@ceph-client ~]# mkdir /data
[root@ceph-client ~]# mount /dev/rbd0 /data/          杰哥的截图水印
[root@ceph-client ~]# cp /etc/passwd /data/
[root@ceph-client ~]# ll /data/
total 4
-rw-r--r-- 1 root root 923 Jun  1 16:33 passwd
[root@ceph-client ~]# df -TH
Filesystem      Type  Size  Used  Avail Use% Mounted on
devtmpfs        devtmpfs 942M    0  942M  0% /dev
tmpfs           tmpfs   954M    0  954M  0% /dev/shm
tmpfs           tmpfs   954M   10M  944M  2% /run
tmpfs           tmpfs   954M    0  954M  0% /sys/fs/cgroup
/dev/mapper/centos-root xfs   129G  1.8G  127G  2% /
/dev/sda1        xfs   534M  196M  338M  37% /boot
tmpfs           tmpfs   191M    0  191M  0% /run/user/0
/dev/rbd0        xfs   1.1G   35M  1.1G  4% /data
[root@ceph-client ~]#
```

4.1.3.6:客户端验证：

```
[root@ceph-client data]# dd if=/dev/zero of=/data/ceph-test-file bs=1MB count=300
300+0 records in
300+0 records out
3000000000 bytes (300 MB) copied, 3.49849 s, 85.8 MB/s
[root@ceph-client data]# file /data/ceph-test-file
/data/ceph-test-file: data

[root@ceph-client ~]# ll -h /data/ceph-test-file
-rw-r--r-- 1 root root 287M Jun  1 16:43 /data/ceph-test-file
```

4.1.3.7:ceph 验证数据:

```
[root@ceph-node2 ~]# ceph df  
GLOBAL:  
  SIZE      AVAIL      RAW USED      %RAW USED  
  1.5 TiB   1.4 TiB    16 GiB       1.07  
POOLS:  
  NAME      ID      USED      %USED      MAX AVAIL      OBJECTS  
  mypool    1        0 B       0          469 GiB       0  
  myrdb1    2        300 MiB   0.06       469 GiB       91  
[root@ceph-node2 ~]#
```

← 已用磁盘空间

4.2: ceph radosgw(RGW)对象存储:

RGW 提供的是 REST 接口，客户端通过 http 与其进行交互，完成数据的增删改查等管理操作。

radosgw 用在需要使用 RESTful API 接口访问 ceph 数据的场合，因此在使用 RBD 即块存储得场合或者使用 cephFS 的场合可以不用启用 radosgw 功能。

4.2.1:部署 radosgw 服务:

如果是在使用 radosgw 的场合，则以下命令将 ceph-mgr1 服务器部署为 RGW 主机：

```
root@ceph-mgr1:~# apt-cache madison radosgw  
root@ceph-mgr1:~# apt install radosgw=16.2.5-1bionic
```

```
[root@ceph-mgr1 ~]# yum install ceph-radosgw  
[ceph@ceph-deploy ceph-cluster]$ ceph-deploy --overwrite-conf rgw create  
ceph-mgr1
```

```
[ceph@ceph-deploy ceph-cluster]$ ceph-deploy rgw create ceph-mgr1  
[ceph_deploy.conf][DEBUG] found configuration file at: /home/ceph/.cephdeploy.conf  
[ceph_deploy.cli][INFO] Invoked (2.0.1): /bin/ceph-deploy rgw create ceph-mgr1  
[ceph_deploy.cli][INFO] ceph-deploy options:  
[ceph_deploy.cli][INFO]   username           : None  
[ceph_deploy.cli][INFO]   verbose            : False  
[ceph_deploy.cli][INFO]   rgw               : [('ceph-mgr1', 'rgw.ceph-mgr1')]  
[ceph_deploy.cli][INFO]   overwrite_conf     : False  
[ceph_deploy.cli][INFO]   subcommand         : create  
[ceph_deploy.cli][INFO]   quiet              : False  
[ceph_deploy.cli][INFO]   cd_conf             : <ceph_deploy.conf.cephdeploy.Conf instance at 0x7fb0d384a3f8>  
[ceph_deploy.cli][INFO]   cluster             : ceph  
[ceph_deploy.cli][INFO]   func                : <function rgw at 0x7fb0d4318050>  
[ceph_deploy.cli][INFO]   ceph_conf           : None  
[ceph_deploy.cli][INFO]   default_release     : False  
[ceph_deploy.rgw][DEBUG] Deploying rgw, cluster ceph hosts ceph-mgr1:rgw.ceph-mgr1  
[ceph-mgr1][DEBUG] connection detected need for sudo  
[ceph-mgr1][DEBUG] connection to host: ceph-mgr1  
[ceph-mgr1][DEBUG] detect platform information from remote host  
[ceph-mgr1][DEBUG] detect machine type  
[ceph_deploy.rgw][INFO] Distro info: CentOS Linux 7.8.2003 Core  
[ceph_deploy.rgw][DEBUG] remote host will use systemd  
[ceph_deploy.rgw][DEBUG] deploying rgw bootstrap to ceph-mgr1  
[ceph-mgr1][DEBUG] write cluster configuration to /etc/ceph/cluster.conf  
[ceph-mgr1][WARNING] rgw keyring does not exist yet, creating one  
[ceph-mgr1][DEBUG] create a keyring file  
[ceph-mgr1][DEBUG] create path recursively if it doesn't exist  
[ceph-mgr1][INFO] Running command: sudo ceph --cluster ceph --name client.bootstrap-rgw --keyring /var/lib/ceph/bootstrap-rgw/ceph.keyring auth get-or-create client.ceph-mgr1 osd allow rwx mon allow rw -o /var/lib/ceph/radosgw/ceph-rgw.ceph-mgr1/keyring  
[ceph-mgr1][INFO] Running command: sudo systemctl enable ceph-radosgw@rgw.ceph-mgr1  
[ceph-mgr1][WARNING] Created symlink from /etc/systemd/system/ceph-radosgw.target.wants/ceph-radosgw@rgw.ceph-mgr1.service to /usr/lib/systemd/system/ceph-radosgw@.service.  
[ceph-mgr1][INFO] Running command: sudo systemctl start ceph-radosgw@rgw.ceph-mgr1  
[ceph-mgr1][INFO] Running command: sudo systemctl enable ceph.target  
[ceph_deploy.rgw][INFO] The Ceph Object Gateway (RGW) is now running on host ceph-mgr1 and default port 7480  
[ceph@ceph-deploy ceph-cluster]$
```

4.2.2:验证 radosgw 服务:

```
[root@ceph-mgr1 ~]# ps -aux | grep radosgw  
ceph          2528  1.5  2.7 5055512 52048 ?          Ssl  16:49   0:00  
/usr/bin/radosgw -f --cluster ceph --name client.rgw.ceph-mgr1 --setuser ceph
```

```
--setgroup ceph
root          3142  0.0  0.0 112808      968 pts/0      S+    16:50   0:00 grep
--color=auto radosgw
```

← → C 不安全 | 172.31.6.104:7480

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<ListAllMyBucketsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Owner>
    <ID>anonymous</ID>
    <DisplayName/>
  </Owner>
  <Buckets/>
</ListAllMyBucketsResult>
```

杰哥的截图水印

4.2.3:验证 ceph 状态:

```
[ceph@ceph-deploy ceph-cluster]$ ceph -s
cluster:
  id:      d7dc1d7a-ac99-4dc0-b551-e92149df7fde
  health:  HEALTH_WARN
           too few PGs per OSD (25 < min 30)          杰哥的截图水印

  services:
    mon: 3 daemons, quorum ceph-mon1,ceph-mon2,ceph-mon3
    mgr: ceph-mgr1(active), standbys: ceph-mgr2
    osd: 15 osds: 15 up, 15 in
    rgw: 1 daemon active

  data:
    pools:   6 pools, 128 pgs
    objects: 278 objects, 300 MiB
    usage:   16 GiB used, 1.4 TiB / 1.5 TiB avail
    pgs:     128 active+clean

[ceph@ceph-deploy ceph-cluster]$
[ceph@ceph-deploy ceph-cluster]$
```

4.2.4:验证 radosgw 存储池:

初始化完成 radosgw 之后，会初始化默认的存储池如下：

```
[ceph@ceph-deploy ceph-cluster]$ ceph osd pool ls
mypool
myrdb1
.rgw.root
default.rgw.control
default.rgw.meta
default.rgw.log
```