Ceph IntroductioN

1. Ceph简介：发展历程| 定义
2. Ceph架构：RADOWS | CRUSH | OSD | MON | MGR | CEPHX | CEPHFS | RBD | RADOSGW
3. Ceph安装：MANUAL | CEPH-DEPLOY | CEPH-DOCKER
4. Ceph性能：RBD

发展历程

2003

Sage Weil在圣克鲁兹加利福尼亚大学时开始开发Ceph系统

2006

在USENIX操作系统设计大会（OSDI 2006）上首次亮相,RADOS

2012

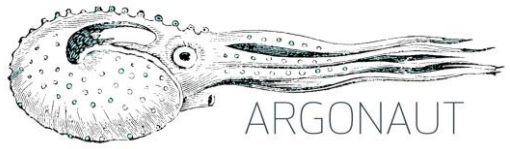
Weil创建了Inktank Storage为Ceph提供支持

04,2014

Red Hat收购了Inktank Storage

Sage Weil

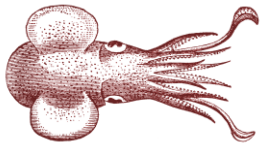
Event Line



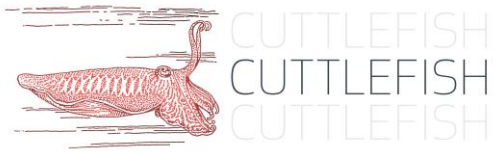
07.03,2012

Release Line

01.01,2013



v0.56



05.07,2013

v0.61

01.01,2013

v0.67

Emperor

11.09,2013

v0.72

05.07,2014

v0.80

10.29,2014

v0.87

04.07,2015

Hammer

v0.94

11.06,2015

v9.2.0

04.21,2016

v10.2.0

01.20,2017

v11.2.0

04.29,2017

v12.2.0

06,2018

v13.2.0

定义

> Ceph is a distributed object store and file system designed to provide excellent performance, reliability and scalability.

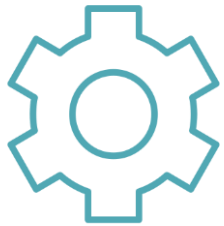
- Ceph是基于RADOS系统构建的，它能在单个的集群环境中提供对象、块设备和文件系统的服务；
- CRUSH算法使Ceph从集中式数据表映射引起的存储集群扩展、性能瓶颈的限制中解脱出来；

对象存储

Ceph提供了兼容S3和Swift的REST风格的接口radowsgw，使得Ceph可以无缝的访问对象存储;

块存储

Ceph的RBD为应用提供了块设备镜像，在集群中镜像被条带化和复制到整个集群;

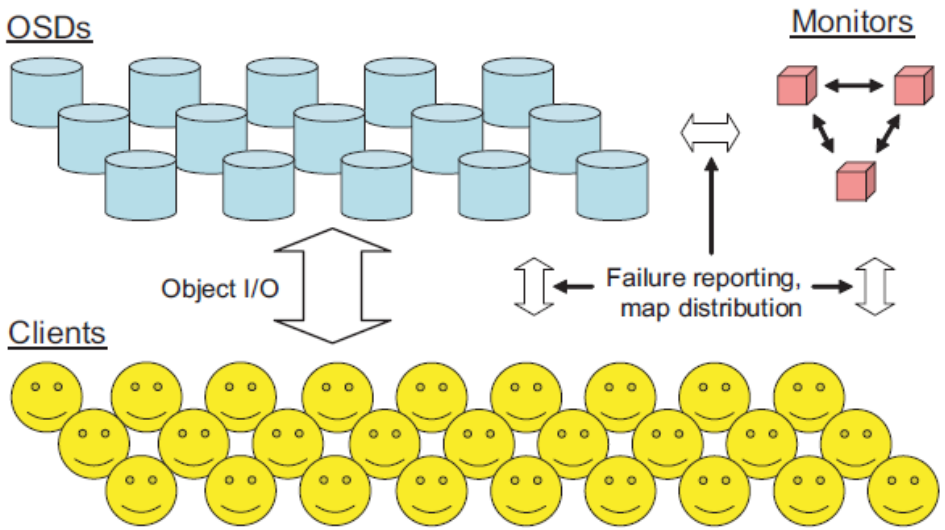


文件系统

Ceph提供高性能和大数据存储、POSIX兼容的网络文件系统CephFS;

RADOWS

RADOS: Reliable,Autonomic Distributed Object Store

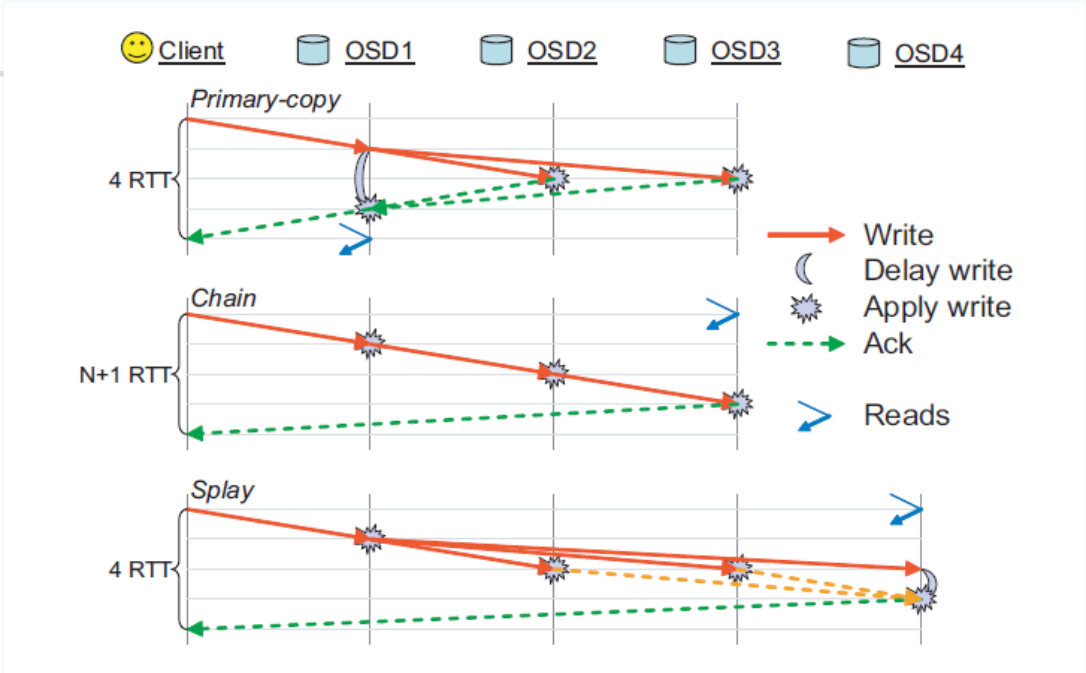


- 数据存储集群OSDs
- 多个Mon组成的集群

集群图	Monitor维护存储单元的关系和状态
数据映射	伪随机定位算法(Hash + CRUSH)
设备状态	in, out, up, down四种状态
Map同步	节点通信，增量惰性同步
Peering	PG内OSD达成相同状态的过程

- 数据迁移
  - 集群扩张或收缩时最小数据迁移
- 失效校验
  - 各节点交换心跳消息
- 数据一致性
  - 一般清洗: 检查对象大小即属性
  - 深度清洗: 检查文件内容

数据复制

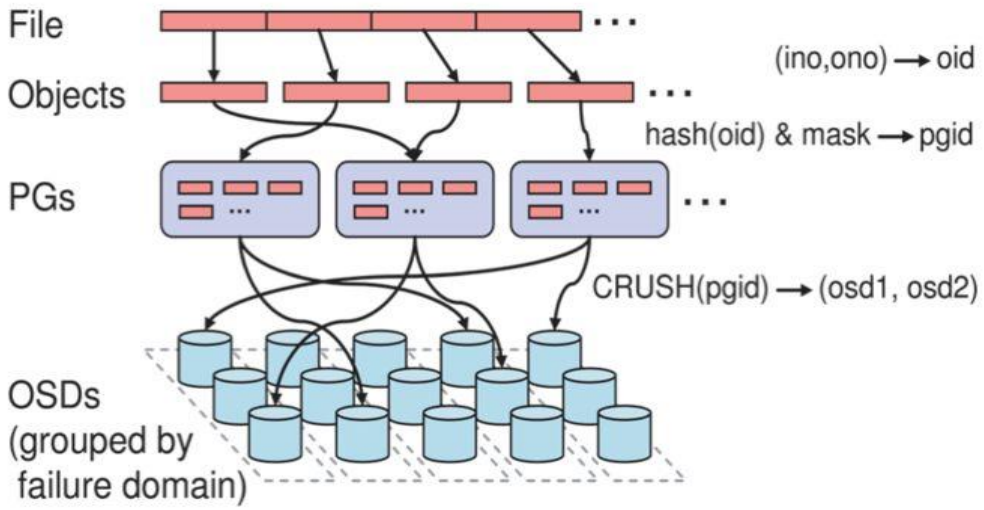

Client
OSD1
OSD2
OSD3
OSD4
Primary-copy
4 RTT
Chain
N+1 RTT
Splay
4 RTT
Write
Delay write
Apply write
Ack
Reads

数据放置

1. File—strip—Objects

2. Hash(Filename + stripNum) % pgNum = PGID
   pgid = PoolID.PGID

3. CRUSH(pgid, r) = [osd.id, osd.id , …]



CRUSH

CRUSH: Controlled Replication Under Scalable Hashing

1. 任何组件都可使用CRUSH
2. 使用Cluster Map
3. 很少的元数据
4. 增强集群性能、可用性和可扩展性

CRUSH MAP

```
# tunable list
tunable choose_local_tries 0
```

```
# devices
device 2 osd.2 class ssd
```

```
# types
type 1 host
```

```
# buckets
host mon { … }
```

```
# rules
rule replicated_rule { … }
```

tuable	算法可调参数部分
devices	集群设备列表，一个设备对应一个OSD
types	Bucket类型列表，表征物理分布
buckets	逻辑拓扑定义
rules	Pool数据放置规则

tunable

tunable {key} {value}: 改进算法的可调整选项

```
choose_total_tries 50                # 选择item时的最大失败次数
chooseleaf_descend_once 1            # 递归算法是否重试
chooseleaf_vary_r 1                  # 递归尝试从非0值r开始
straw_calc_version 1                 # 用于修复straw算法
```

类型修改

ceph osd crush tunables
legacy|argonaut|bobtail|firefly|hammer|optimal|default

！ Note: 修改tunables类型会导致集群数据的迁移

types

```
type 1 host
type 2 chassis
type 3 rack
type 4 row
type 5 pdu
type 6 pod
type 7 room
type 8 datacenter
type 9 region
type 10 root
```

roles

```
rule <rulename> {
        ruleset <ruleset>
        type [ replicated | erasure ]
        min_size <min-size>
        max_size <max-size>
        step take <bucket-name>
        step [choose|chooseleaf] [firstn|
indep] <N> <bucket-type>
        step emit
}
```

buckets

```
[bucket-type] [bucket-name] {
        id [a unique negative numeric ID]
        weight [the relative capacity/capability of the item(s)]
        alg [the bucket type: uniform | list | tree | straw  | straw2]
        hash [the hash type: 0 by default]
        item [item-name] weight [weight]
}
```

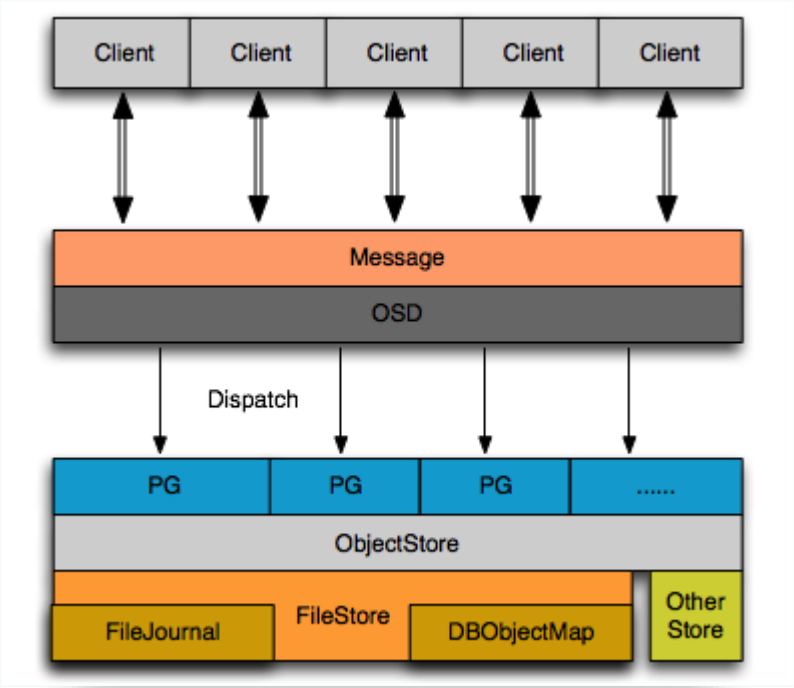
uniform	O(1) 无视权重；数据完全重组
list	O(n) 增加节点时最优
tree	O(log n) 大量节点
straw	O(n) 删除、增加最优
straw2	O(n) 删除、增加最优

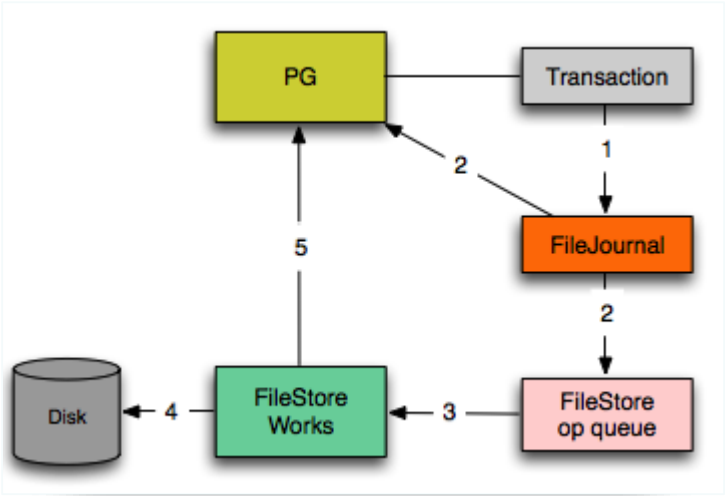
获取CRUSH MAP	ceph osd getcrush –o crush.dump
反编译CRUSH MAP	crushtool -d crush.dump –o crush.txt
编译CRUSH MAP	crushtool –c crush.txt –o crush.dump
注入CRUSH MAP	ceph osd setcrushmap –i crush.dump
设置规则	ceph osd pool set <pool-name> crush_rule <rule-name>

OSD

！ Note: object属性会使用文件系统xattr属性存取

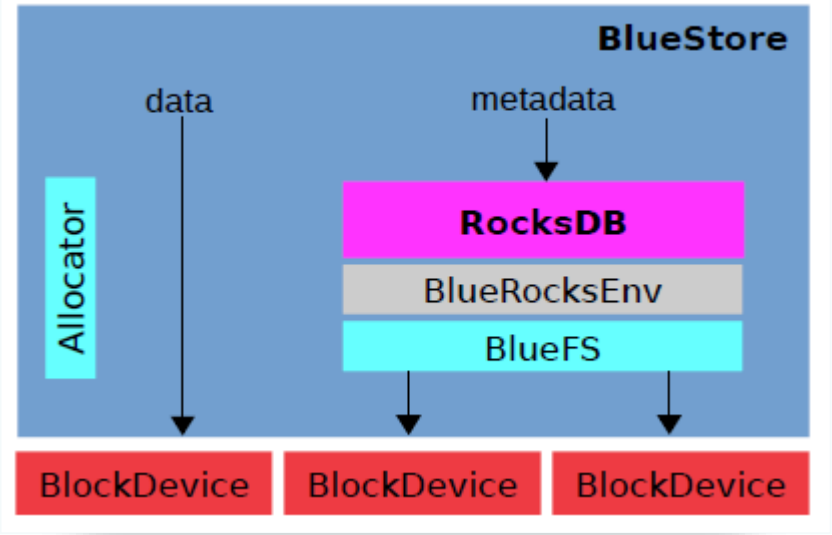
！ Note: 超出的元数据保存在ObjectMap里，即omap;omap部分即key-value DB LevelDB,RocksDB





- 将对象作为文件保存在数据目录下

- filestore WAL存在一倍写放大问题

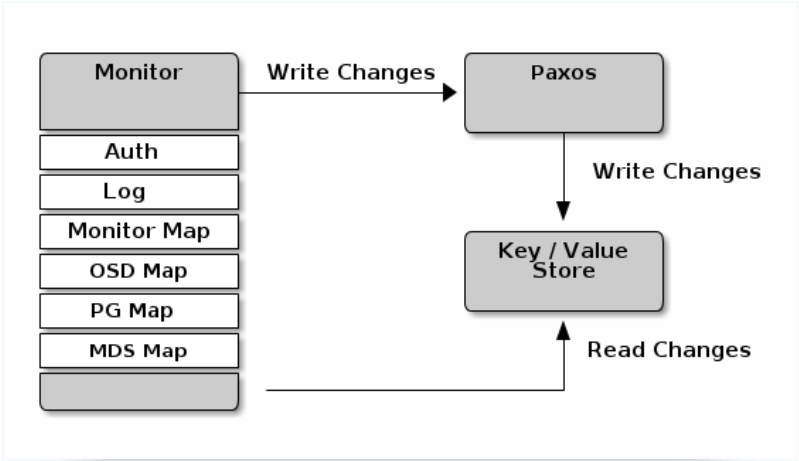


- 直接管理裸设备
- 针对SSD进行了优化
- 解决了filestore数据落盘问题

RocksDB	对象元数据、omap数据信息以及分配器元数据
BlueRocksEnv	与RocksDB交互的接口
BlueFS	实现RockesEnv的接口

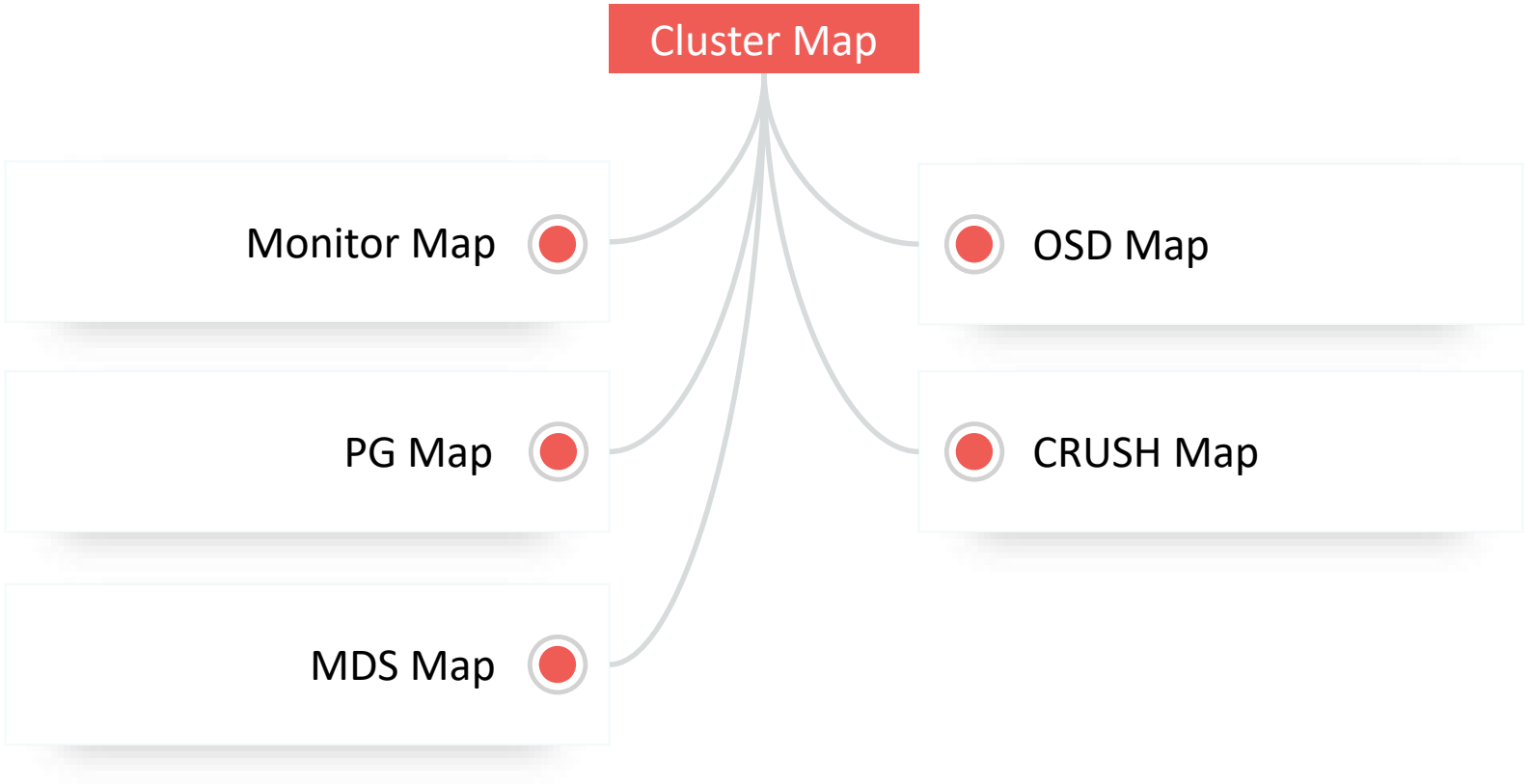
```
ceph-volume lvm create --bluestore --data { device } --block.db { device } --block.wal
{ device }
```

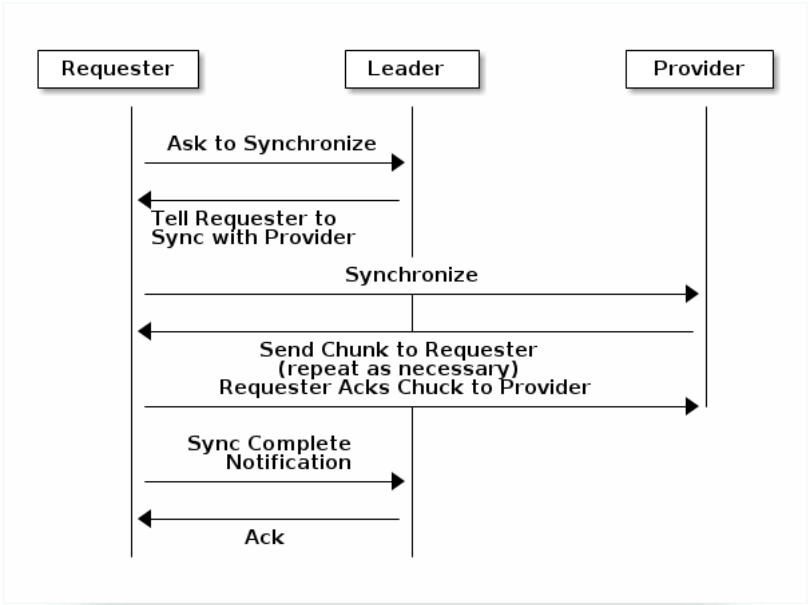
MON



- 保存集群图主副本
- CephX认证服务
- 日志服务

！Note: monitor和monitor通过monmap来通信，而不是配置文件


Cluster Map
Monitor Map
OSD Map
PG Map
CRUSH Map
MDS Map



Leader	最先生成Cluster Map,并选举成功
Provider	拥有最新的Cluster Map
Requester	非Quorum成员

MGR

描述

为外部管理系统提供Ceph集群监控接口

新增

mkdir /var/lib/ceph/mgr/ceph-{ id }/

ceph auth get-or-create mgr.{ id } mon ‘allow profile mgr’ osd ‘allow *’ mds ‘allow *’ > /var/lib/ceph/mgr/ceph-{ id }/keyring

sudo ceph-mgr -i { id } --setuser ceph --setgroup ceph

sudo systemctl start ceph-mgr@{ id }

删除

sudo systemctl disable ceph-mgr@{ id }

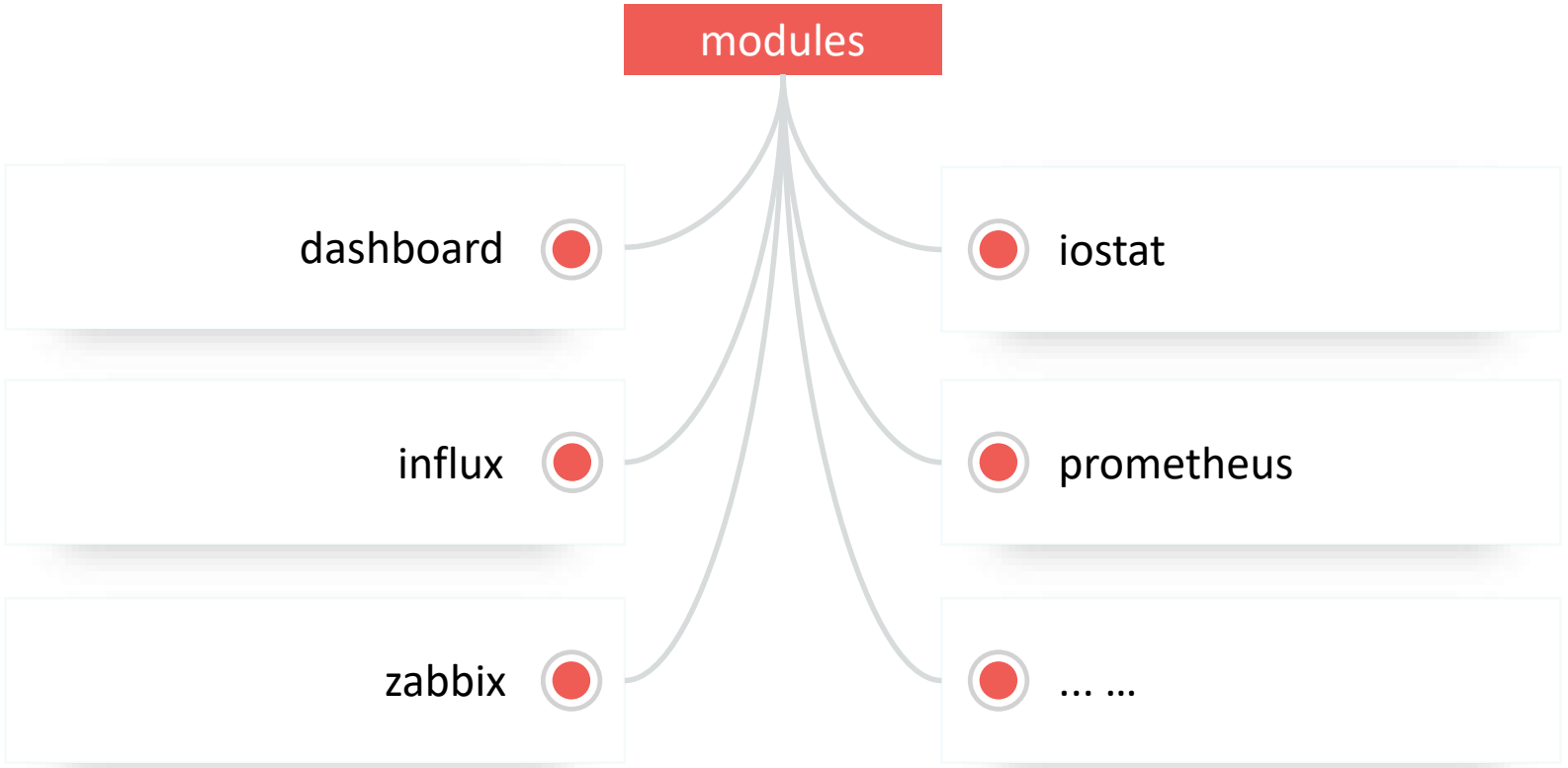
sudo systemctl stop ceph-mgr@{ id }

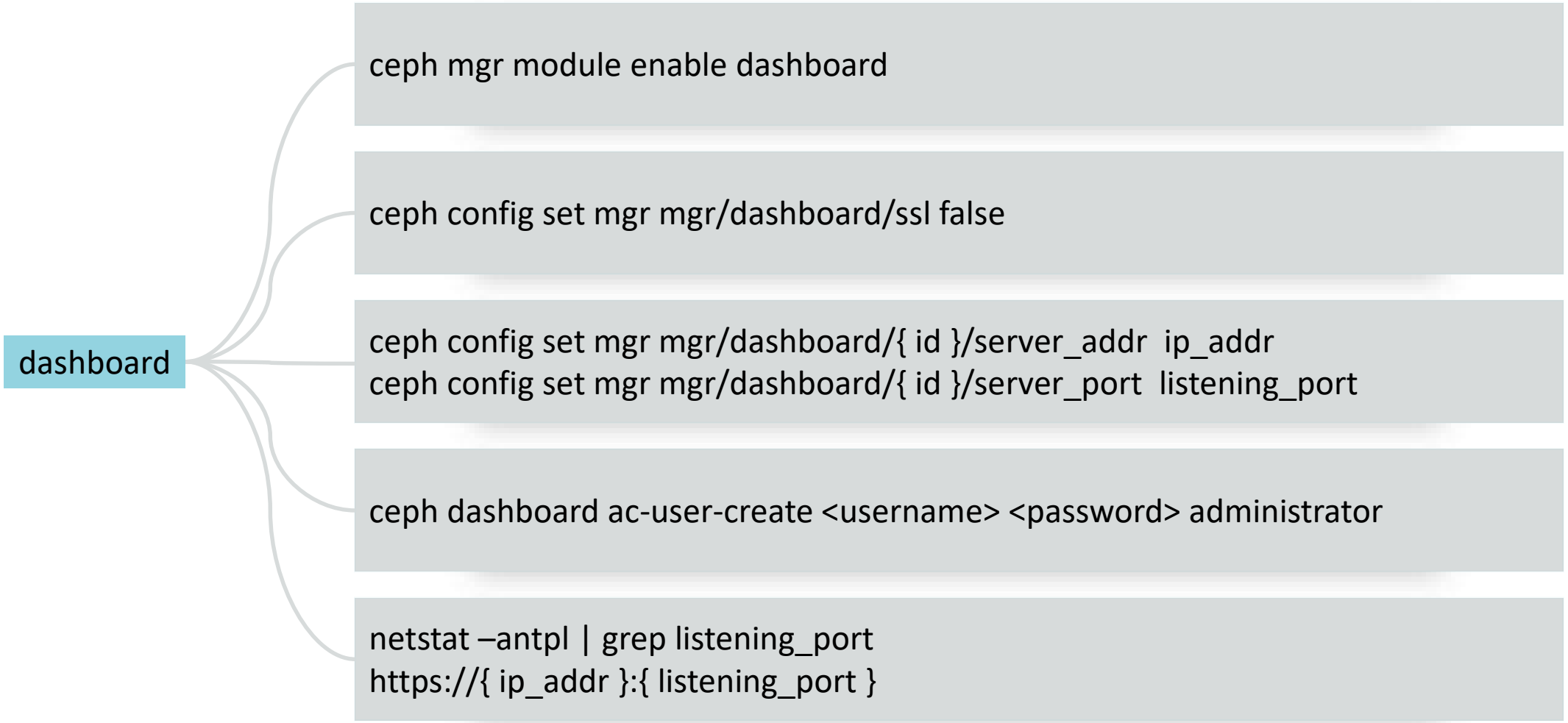
ceph auth del mgr.{ id }
rm -f /var/lib/ceph/mgr/ceph-{ id }/keyring

edit /etc/ceph/ceph.conf

查询

ceph mgr module ls




dashboard
ceph mgr module enable dashboard
ceph config set mgr mgr/dashboard/ssl false
ceph config set mgr mgr/dashboard/{ id }/server_addr ip_addr
ceph config set mgr mgr/dashboard/{ id }/server_port listening_port
ceph dashboard ac-user-create <username> <password> administrator
netstat –antpl | grep listening_port
https://{ ip_addr }:{ listening_port }

CEPHX

< TYPE. ID>

- mon: ID为空
- osd: ID为OSD的id
- client: ID为客户端名称

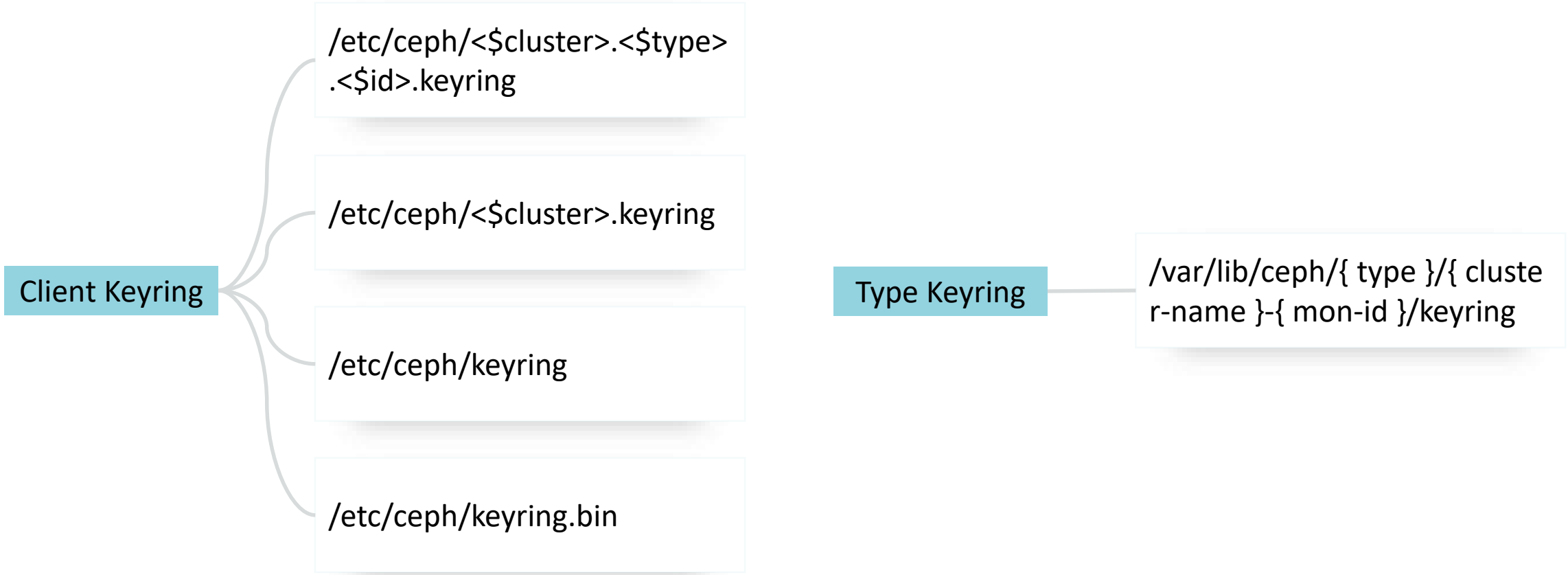
```
ceph -s --conf /etc/ceph/ceph.conf --name client.admin --keyring
/etc/ceph/ceph.client.admin.keyring
```

```
# 集群进程间消息认证
auth cluster required = cephx

# 客户端到集群服务的认证
auth service required = cephx

# 集群到客户端的认证
auth client required = cephx
```

！ Note: monitor的密钥修改后不影响集群


Client Keyring
/etc/ceph/<$cluster>.<$type>.<$id>.keyring
/etc/ceph/<$cluster>.keyring
/etc/ceph/keyring
/etc/ceph/keyring.bin
Type Keyring
/var/lib/ceph/{ type }/{ cluster-name }-{ mon-id }/keyring

caps： type priv

ceph auth get-or-create type.{ id } caps

mon

- r：获取Map数据，集群状态
- w：修改Osd状态，修复PG，修改Map,删除Mon
- x：执行auth命令相关

profile osd：给用户一个osd的身份

！ *Note: * = rwx*

！ *Note: w和x权限都必须搭配r*

命令

ceph auth list
ceph auth export {<entity>}

ceph auth add <entity> {<caps> [<caps>...]}
ceph auth caps <entity> <caps> [<caps>...]

ceph auth del <entity>

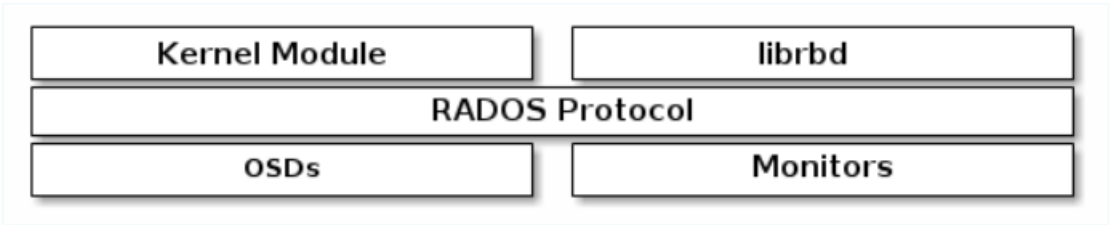
ceph auth get <entity>
ceph auth get-key <entity>

ceph auth get-or-create <entity> {<caps> [<caps>...]}
ceph auth import

RBD

描述

Ceph集群提供的块设备



- kernel Module
- librbd

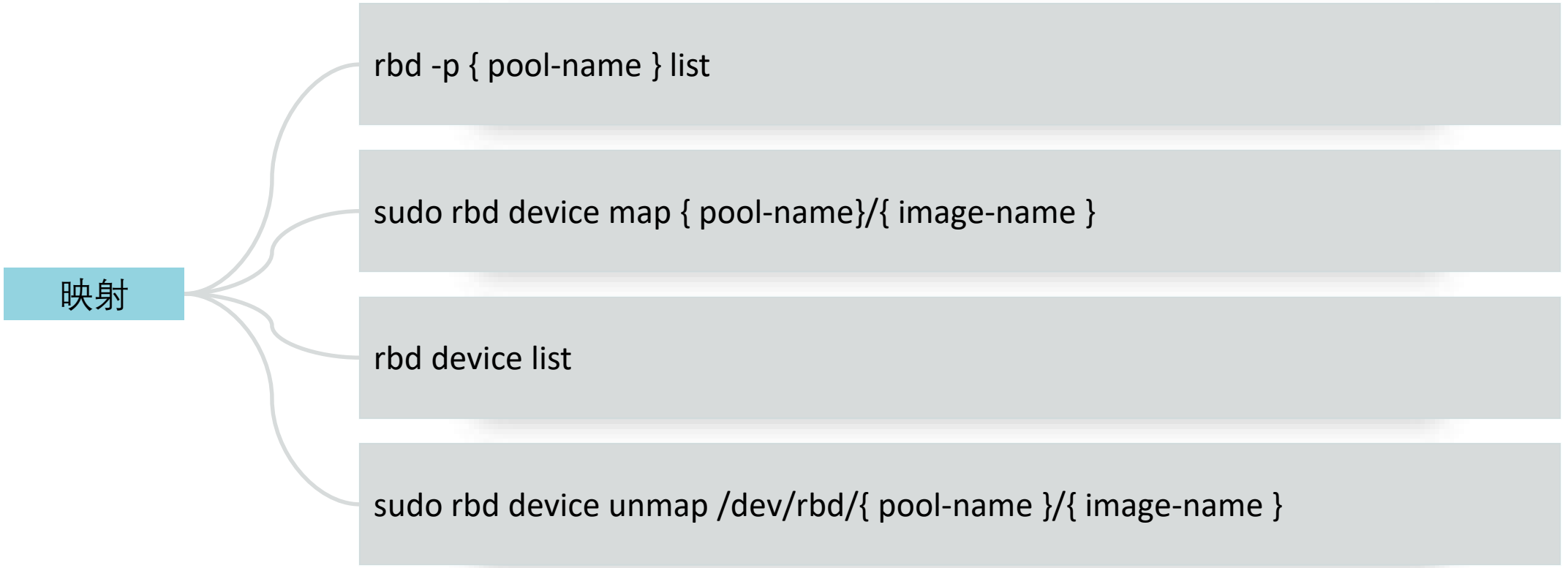
创建

ceph osd pool create { pool-name } <pg_num> <pgp_num>

rbd pool init { pool-name }

rbd create --size { megabytes } { pool-name }/{ image-name }

rbd ls { pool-name }


映射
rbd -p { pool-name } list
sudo rbd device map { pool-name}/{ image-name }
rbd device list
sudo rbd device unmap /dev/rbd/{ pool-name }/{ image-name }

删除

- sudo rbd device unmap /dev/rbd/{ pool-name }/{ image-name }
- rbd rm { pool-name }/{ image-name }
- rbd trash mv { pool-name }/{ image-name}
- rbd trash rm { pool-name }/{ image-name }

快照

特定时间点块设备的只读拷贝

命令

rbd snap create { pool-name }/{ image-name }@{ snap-name }
rbd snap rm { pool-name}/{ image-name }@{ snap-name }

rbd snap ls { pool-name }/{ image-name }

rbd snap roolback { pool-name }/{ image-name }

rbd snap purge { pool-name }/{ image-name }

！ Note: 做快照时需要关闭块设备IO(fsfreeze)

克隆

以快照为基础的可写块设备

命令

rbd protect { pool-name }/{ image-name }
rbd unprotect { pool-name }/{ image-name }

rbd clone { pool-name }/{ image-name }@{ snap-name } { pool-name }/{ image-name}

rbd children { pool-name }/{ image-name}@{ snap-name }

rbd flatten { pool-name }/{ image-name }

！ Note: 删除有克隆的快照会导致数据丢失

！ Note: protect的快照不能删除

镜像

块设备或存储次的冗余副本

命令

rbd mirror pool enable { pool-name } { mode } # mode = pool/image
rbd mirror pool disable { pool-name } { mode }

rbd mirror pool peer add { pool-name } { client-name }@{ cluster-name }

rbd mirror pool info { pool-name }
rbd mirror pool peer remove { pool-name } { peer-uuid }

！ Note: 运行Ceph的镜像功能必须有两个集群

CEPHFS

描述

是兼容POSIX的文件系统

新增

ceph osd pool create cephfs_fs <pg_num> <pgp_num>

ceph osd pool create cephfs_mt <pg_num> <pgp_num>

ceph fs new <fs-name> <mt_pool> <fs_pool>

ceph fs ls

！ Note: CephFS需要使用MDS服务

挂载

```
lsmod | grep rbd
# 从ceph.client.admin.keyring中获取admin用户的keyring
```

```
# 配置挂载点
sudo mount -t ceph {mds-ip}:6789:/ /mount-point -o name=admin,secret=keyring
```

删除

```
sudo umount /mount-point
sudo systemctl stop ceph-mds@{ mds-id }
```

```
ceph fs rm { fs-name } --yes-i-really-mean-it
```

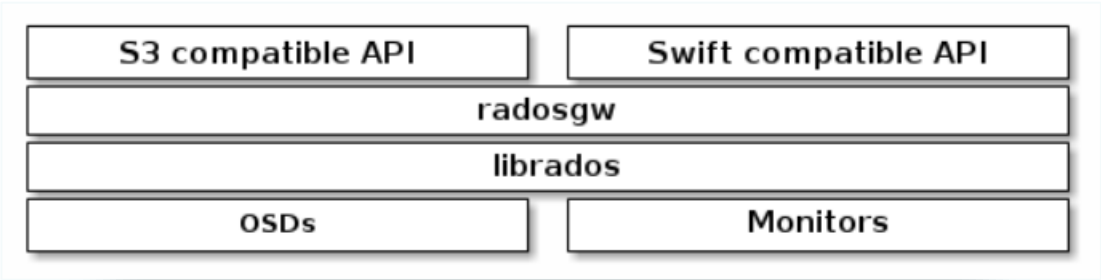
```
ceph osd pool cephfs_fs cephfs_fs --yes-i-really-really-mean-it
ceph osd pool cephfs_mt cephfs_mt --yes-i-really-really-mean-it
```

！ Note: CephFS使用ceph fs set-default设置默认文件系统

RGW

描述

构建于libradows库上的RESTful风格的对象存储接口



- 兼容S3 API
- 兼容Swift API

MANUAL INSTALL

命令

1. 配置Ceph源 安装ceph
   yum -y install ceph ceph-radosgw
2. mkdir /var/lib/ceph/mon/{ cluster-name }-{ mon-id }
   sudo touch /etc/ceph/ceph.conf
3. uuid
   编辑配置文件ceph.conf
4. ceph-authtool --create-keyring /tmp/ceph.mon.keyring --gen-key -n mon. --cap mon 'allow *'
5. ceph-authtool --create-keyring /etc/ceph/ceph.client.admin.keyring --gen-key -n client.admin --set-uid=0 --cap mon 'allow *' --cap osd 'allow *' --cap mds 'allow *' --cap mgr 'allow *'
6. ceph-authtool /tmp/ceph.mon.keyring --import-keyring /etc/ceph/ceph.client.admin.keyring

命令

7. monmaptool --create --add $name $ip --fsid $uuid /tmp/monmap

8. ceph-mon --mkfs -i $name --monmap /tmp/monmap –keyring /tmp/ceph.mon.keyring

9. sudo touch /var/lib/ceph/mon/{ cluster-name }-{ mon-id }/done

1. sudo systemctl start ceph-mon@{ mon-id }
   ceph -s

2. mkdir /var/lib/ceph/bootstrap-osd/
   mkdir /var/lib/ceph/osd/{ cluster-name }-{ osd-id }

3. cp mon@ceph.conf /etc/ceph/ceph.conf

命令

4. ceph-authtool --create-keyring /var/lib/ceph/bootstrap-osd/ceph.keyring --gen-key -n client.bootstrap-osd --cap mon 'profile bootstrap-osd'

5. Bluestore:
sudo ceph-volume lvm create --bluestore --data { device } --block.wal { device } --block.db { device }

6. Filestore:
ceph-volume lvm create --filestore --data { data lv} --journal { journal device }

7. sudo systemctl enable ceph-osd@{ osd-id }
sudo systemctl start ceph-osd@{ osd-id }

CEPH-DEPLOY

1. 创建安装用户
   配置sudo用户无密码
2. ntp下载配置
   python安装
3. admin节点hosts解析配置
   ssh无密登陆配置
4. 关闭防火墙
   OSD节点磁盘配置
5. 下载ceph-deploy
   确定安装目录，初始化集群
6. 编辑配置文件
   为各节点安装Ceph

7 初始化Monitor
准备OSD磁盘

8 创建OSD
分发admin用户的keyring

9 查看集群状态

命令

1. ceph-deploy new [-h] [--no-ssh-copykey] [--fsid FSID] [--cluster-network CLUSTER_NETWORK] [--public-network PUBLIC_NETWORK] MON [MON ...]
2. ceph-deploy mon create-initial # mon initial members
   ceph-deploy mon create nodename
3. ceph-deploy osd create {node} --data /path/to/data --block-db /path/to/db-device --block-wal /path/to/wal-device
4. ceph-deploy mgr create nodename
   ceph-deploy mds create nodename
5. ceph-deploy rgw create nodename
   ceph-deploy uninstall nodename
6. ceph-deploy purgedata nodename
   ceph-deploy purge nodename

CEPH-CONTAINER

描述

编译ceph docker镜像

编译启动

```
git clone https://github.com/ceph/ceph-container
make FLAVORS=mimic,centos,7 build
```

```
docker run -d --net=host -v /etc/ceph:/etc/ceph  -v /var/lib/ceph/:/var/lib/ceph/ -e MON_IP=IP_ADDR -e CEPH_PUBLIC_NETWORK=IP_ADDR ceph/daemon mon
```

```
docker run -d --net=host --privileged=true --pid=host -v /etc/ceph:/etc/ceph -v /var/lib/ceph/:/var/lib/ceph/ -v /dev/:/dev/ -e OSD_DEVICE=/dev/vdd -e OSD_TYPE=disk -e OSD_BLUESTORE=1 ceph/daemon osd
```

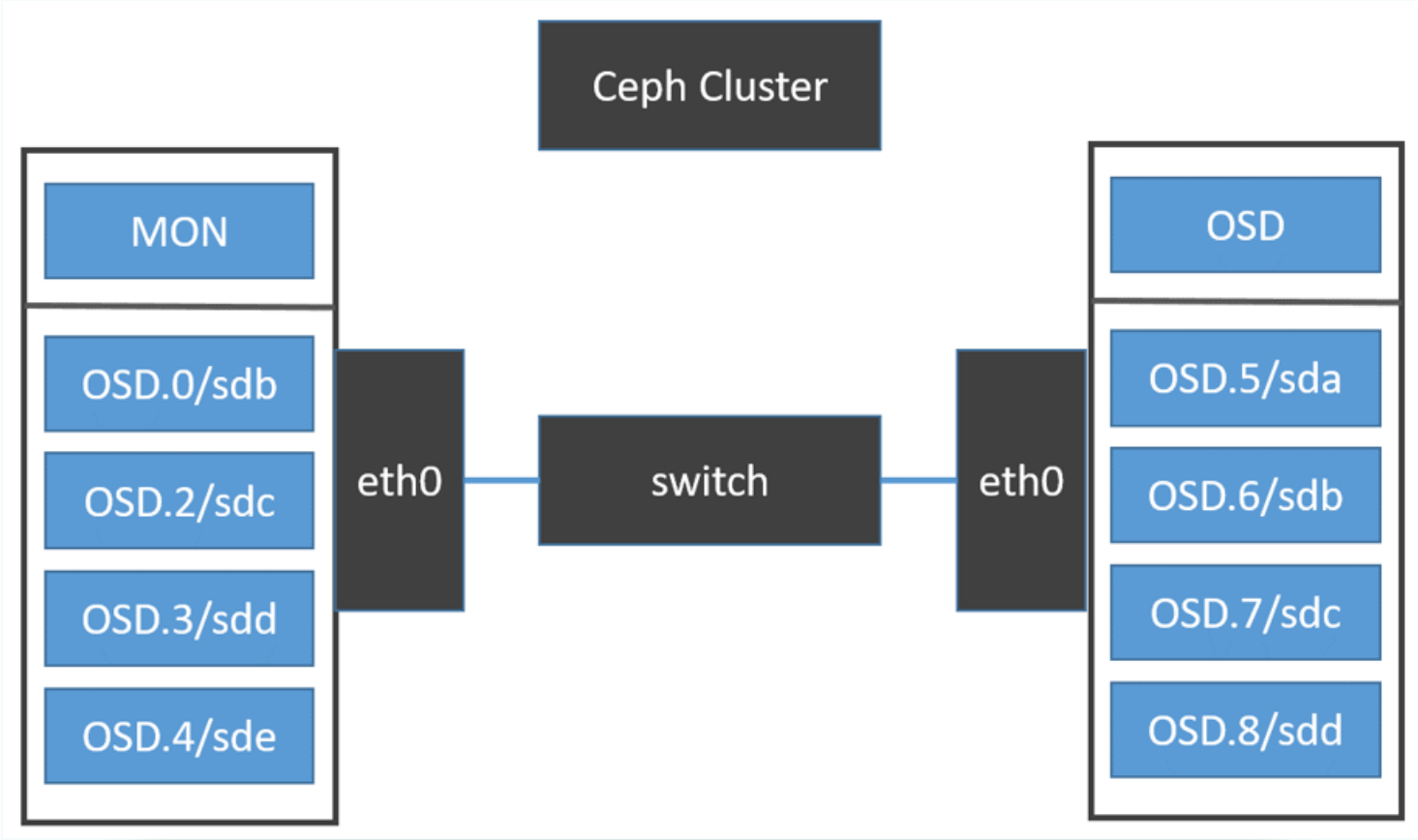
```
docker run -d --net=host -v /etc/ceph:/etc/ceph -v /var/lib/ceph/:/var/lib/ceph/ ceph/daemon mgr
```

！ Note: 系统上需要安装docker

RBD性能

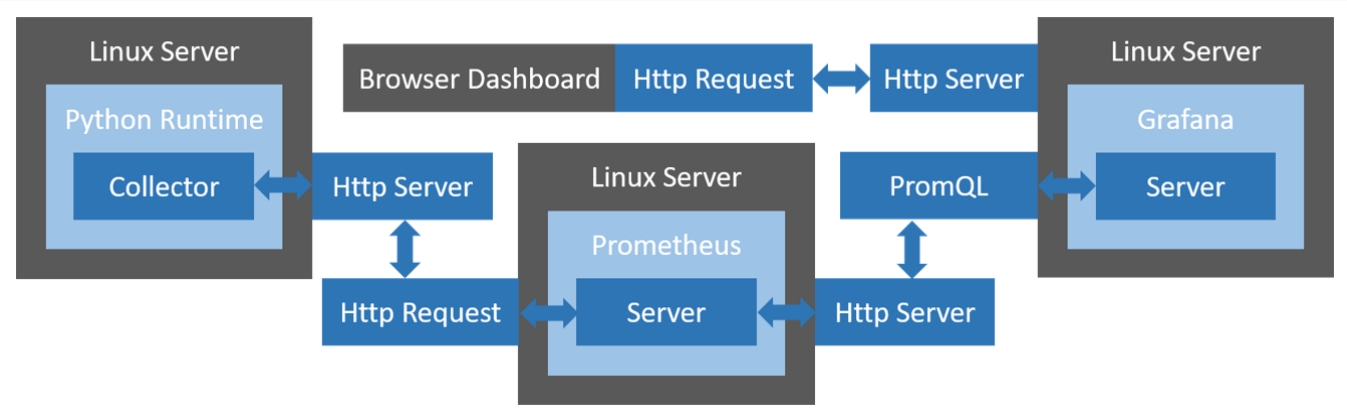
原理

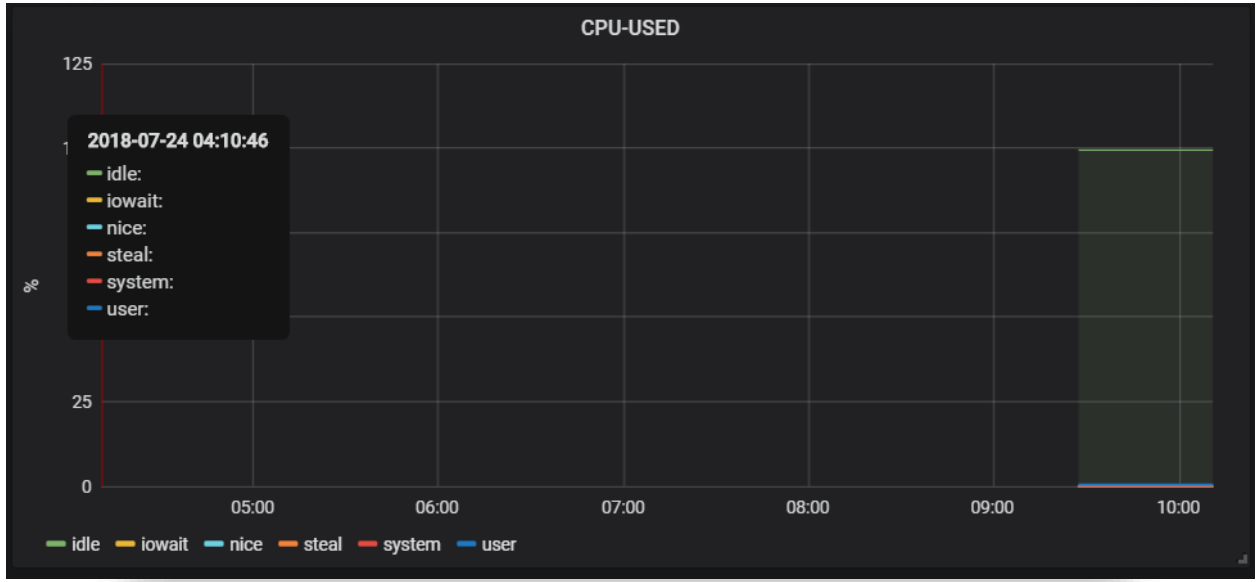
blktrace + fio + ceph RBD



附录

Exporter + Prometheus + Grafana





Q & A