Ceph故障运维手册

目录

[Monitor 2](#_heading=h.khcg4ws3ev32)

[删除monitor 2](#_heading=h.3svl0gjkfomk)

[使用ceph-ansible删除monitor 2](#_heading=h.vmc7tdf9jqpl)

[手动删除monitor 3](#_heading=h.oeb3wlo609z8)

[添加monitor 3](#_heading=h.xvpe5ivuijlg)

[使用ceph-ansible添加monitor 3](#_heading=h.g2weok9he37h)

[手动添加monitor 4](#_heading=h.tmd569k4z9rh)

[mons are using a lot of disk space 5](#_heading=h.flt59lz46cm3)

[mon low disk space 5](#_heading=h.xbgcn1q8bfmy)

[osd 5](#_heading=h.ejxcgcyntnhb)

[删除osd 6](#_heading=h.3fybt9cxcypl)

[使用ceph-ansible删除osd 6](#_heading=h.458dahjr6c6)

[手动删除osd 6](#_heading=h.u4nqrk3dx1io)

[添加osd 7](#_heading=h.qhv0ol1wwwxv)

[使用ceph-ansible 添加osd 7](#_heading=h.rm11efxk8ma6)

[手动添加osd 8](#_heading=h.7b4rvnwvo6i0)

[处理osd down故障 8](#_heading=h.y3pggt25mkj8)

[pg故障 11](#_heading=h.igznuvwd0i7t)

[pg inconsistent 11](#_heading=h.teqecjk1kj77)

[pg incomplete 11](#_heading=h.c9nzhpac0ukk)

[pg stale 12](#_heading=h.z7cjrebwwj24)

[pg remapped 12](#_heading=h.lvug0xk2j2lq)

[pg down 12](#_heading=h.qfka1hgiauie)

[pg degraded+peering/recovering 14](#_heading=h.j9tsf1sk9cat)

[object故障 14](#_heading=h.ozdwmxex6hy3)

[slow requests 17](#_heading=h.ickmt75ms8z1)

# Monitor

## 删除monitor

### 使用ceph-ansible删除monitor

1.切换到ceph-ansible目录/usr/share/ceph-ansible/

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

2.拷贝infrastructure-playbooks/shrink-mon.yml文件到当前目录

| [root@admin ceph-ansible]# cp infrastructure-playbooks/shrink-mon.yml . |
| --- |

3.执行shrink-mon.yml playbook

| [user@admin ceph-ansible]$ ansible-playbook shrink-mon.yml -e mon\_to\_kill=<hostname> -u <ansible-user> |
| --- |

替换 <hostname> 为monitor的主机名

替换 <ansible-user> 为执行ceph-ansible的用户，如果使用root用户执行，这个参数可以不写。

例子：

| [user@admin ceph-ansible]$ ansible-playbook shrink-mon.yml -e mon\_to\_kill=monitor1 -u user |
| --- |

4.删除集群所有节点/etc/ceph/ceph.conf中被删除的monitor相关内容

| mon host = 10.72.44.155:6789,10.72.44.156:6789,10.72.44.157:6789 |
| --- |

5.确认monitor是否删除成功

| [root@monitor ~]# ceph -s |
| --- |

### 手动删除monitor

1.停止需要删除的monitor服务

| systemctl stop ceph-mon@$MONITOR\_ID  例子  [root@monitor ~]# systemctl stop ceph-mon@node3 |
| --- |

2.从集群中删除monitor

| ceph mon remove $MONITOR\_ID  例子  [root@monitor ~]# ceph mon remove node3 |
| --- |

3.删除集群所有节点/etc/ceph/ceph.conf中被删除的monitor相关内容

| mon host = 10.72.44.155:6789,10.72.44.156:6789,10.72.44.157:6789 |
| --- |

1. 备份或者删除已删除monitor相关的数据

| 备份  mv /var/lib/ceph/mon/$CLUSTER\_NAME-$MONITOR\_ID /var/lib/ceph/mon/removed-$CLUSTER\_NAME-$MONITOR\_ID  例子  [root@monitor ~]# mv /var/lib/ceph/mon/ceph-node3 /var/lib/ceph/mon/removed-ceph-node3 |
| --- |

| 删除  rm -r /var/lib/ceph/mon/$CLUSTER\_NAME-$MONITOR\_ID  例子  [root@monitor ~]# rm -r /var/lib/ceph/mon/ceph-node3 |
| --- |

## 添加monitor

### 使用ceph-ansible添加monitor

1.在Ansible inventory文件中的[mon]节中添加新增monitor的相关信息

| [mons]  monitor01  monitor02  monitor03  $NEW\_MONITOR\_NODE\_NAME  $NEW\_MONITOR\_NODE\_NAME |
| --- |

2.验证ansible是否能访问所有节点

| # ansible all -m ping |
| --- |

3.切换到ceph-ansible执行目录

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

4.执行ceph-ansible playbook

| $ ansible-playbook site.yml |
| --- |

### 手动添加monitor

1.在所有集群节点的配置文件/etc/ceph/ceph.conf添加新增monitor的ip

| [mon]  mon host = 192.168.0.1:6789 192.168.0.2:6789 192.168.0.3:6789 192.168.0.4:6789 |
| --- |

2.在新增的monitor节点的服务器上操作，创建monitor的目录

| mkdir /var/lib/ceph/mon/$CLUSTER\_NAME-$MONITOR\_ID  例子  [root@monitor ~]# mkdir /var/lib/ceph/mon/ceph-node4 |
| --- |

3.在新增的monitor节点和活跃的monitor节点上创建临时目录，用来收集新增monitor的信息文件

| [root@monitor ~]# mkdir /tmp/ceph |
| --- |

4.拷贝admin keyring 到新的monitor节点

| scp /etc/ceph/ceph.client.admin.keyring node4:/etc/ceph |
| --- |

5.在活跃的monitor节点上收集mon keyring和mon map

| ceph auth get mon. -o /tmp/ceph/ceph\_keyring.out  ceph mon getmap -o /tmp/ceph/ceph\_mon\_map.out |
| --- |

6.在活跃monitor节点上收集的mon信息拷贝到新增的monitor节点

| scp /tmp/ceph node4:/tmp/ceph |
| --- |

7.使用采集的信息创建monitor

| ceph-mon -i $MONITOR\_ID --mkfs --monmap /$TEMP\_DIRECTORY/$MONITOR\_MAP\_FILE --keyring /$TEMP\_DIRECTORY/$KEY\_FILE\_NAME  例子  [root@monitor ~]# ceph-mon -i node4 --mkfs --monmap /tmp/ceph/ceph\_mon\_map.out --keyring /tmp/ceph/ceph\_keyring.out |
| --- |

1. 提升monitor信息目录的权限

| [root@monitor ~]# chown -R ceph:ceph /var/lib/ceph/mon  [root@monitor ~]# chown -R ceph:ceph /var/log/ceph  [root@monitor ~]# chown -R ceph:ceph /var/run/ceph  [root@monitor ~]# chown -R ceph:ceph /etc/ceph |
| --- |

9.启动新增的monitor服务

| systemctl enable ceph-mon.target  systemctl enable ceph-mon@$MONITOR\_ID  systemctl start ceph-mon@$MONITOR\_ID |
| --- |

## mons are using a lot of disk space

mon在短期内产生的osd map过多，没有删除，导致mon的数据库文件变大，SST文件占用了很大的空间，这时候需要手动压缩数据库文件

ceph tell mon.<host-name> compact

## mon low disk space

mon 的数据存放目录，空间不够使用

需要重新调整系统的分区，扩大mon目录的分区。

例子如下

| /dev/mapper/centos-var 50G 35G 16G 70% /  /dev/mapper/centos-home 3.6T 33M 3.6T 1% /home |
| --- |

可以缩小/home目录的分区，扩大/var目录的分区

# osd

## 删除osd

### 使用ceph-ansible删除osd

1.切换到/usr/share/ceph-ansible/目录

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

1. 拷贝infrastructure-playbooks/shrink-osd.yml到当前目录

| [root@admin ceph-ansible]# cp infrastructure-playbooks/shrink-osd.yml . |
| --- |

3.执行ceph-ansible playbook

| ansible-playbook shrink-osd.yml -e osd\_to\_kill=$ID -u $ANSIBLE\_USER  例子  [user@admin ceph-ansible]$ ansible-playbook shrink-osd.yml -e osd\_to\_kill=1 -u user |
| --- |

4.验证osd成功删除

| ceph osd tree |
| --- |

### 手动删除osd

| ceph osd out <osd\_id>  systemctl stop [ceph-osd@<osd\_id>](about:blank)  ceph osd crush remove osd.<osd\_id>  ceph auth del osd.<osd\_id>  ceph osd rm <osd\_id> |
| --- |

例子

| ceph osd out 4  systemctl stop [ceph-osd@4](about:blank)  ceph osd crush remove osd.4  ceph auth del osd.4  ceph osd rm 4 |
| --- |

删除osd后后续复用osd id

| ceph osd destroy osd.id --yes-i-really-mean-it |
| --- |

## 添加osd

### 使用ceph-ansible 添加osd

1.修改ceph-ansible的inventory文件, 在/etc/ansible/hosts的[osds]节中添加新osd的服务器主机名，如果已有主机名则不添加

| [osds]  ...  osd06  $NEW\_OSD\_NODE\_NAME |
| --- |

2.按照/usr/share/ceph-ansible/group\_vars/osds.yml对新增的osd硬盘做逻辑卷

| osd\_objectstore: bluestore  osd\_scenario: lvm  lvm\_volumes:  - data: data-lv1  data\_vg: data-vg1  db: db-lv1  db\_vg: db-vg1  - data: data-lv2  data\_vg: data-vg2  db: db-lv2  db\_vg: db-vg2 |
| --- |

lvm创建例子如下

| LV VG  lv-ceph-block-1-1 vg-ceph-block-1  lv-ceph-block-1-2 vg-ceph-block-1  lv-ceph-block-1-3 vg-ceph-block-1  lv-ceph-block-1-4 vg-ceph-block-1  lv-ceph-block-2-1 vg-ceph-block-2  lv-ceph-osd-1 vg-ceph-osd-1  lv-ceph-osd-10 vg-ceph-osd-10  lv-ceph-osd-11 vg-ceph-osd-11  lv-ceph-osd-12 vg-ceph-osd-12  lv-ceph-osd-13 vg-ceph-osd-13 |
| --- |

验证ansible是否能访问ceph集群每一个节点

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

切换到/usr/share/ceph-ansible目录

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

拷贝add-osd.yml到/usr/share/ceph-ansible

| [user@admin ceph-ansible]$ sudo cp infrastructure-playbooks/add-osd.yml . |
| --- |

执行ansible脚本

| [user@admin ceph-ansible]$ ansible-playbook add-osd.yml |
| --- |

### 手动添加osd

1.新增的硬盘做逻辑卷

| LV VG  lv-ceph-block-1-1 vg-ceph-block-1  lv-ceph-block-1-2 vg-ceph-block-1  lv-ceph-block-1-3 vg-ceph-block-1  lv-ceph-block-1-4 vg-ceph-block-1  lv-ceph-block-2-1 vg-ceph-block-2  lv-ceph-osd-1 vg-ceph-osd-1  lv-ceph-osd-10 vg-ceph-osd-10  lv-ceph-osd-11 vg-ceph-osd-11  lv-ceph-osd-12 vg-ceph-osd-12  lv-ceph-osd-13 vg-ceph-osd-13 |
| --- |

2.创建osd

| ceph-volume lvm create --bluestore --data VOLUME\_GROUP/LOGICAL\_VOLUME  例子  [root@osd ~]# ceph-volume lvm create --bluestore --data example\_vg/data\_lv |
| --- |

## 处理osd down故障

1.检查故障的osd

查看集群的健康情况

| # ceph health |
| --- |

查看down的osd

| # ceph osd tree | grep -i down |
| --- |

2.尝试启动down的osd

| systemctl start ceph-osd@$OSD\_ID |
| --- |

3.如果osd重启失败

查看/var/log/message和/var/log/ceph/ceph-osd.x.log

日志中显示 IO error

| [24934098.161918] Buffer I/O error on device sdaj, logical block 4  [24934098.161921] Buffer I/O error on device sdaj, logical block 5  [24934098.161922] Buffer I/O error on device sdaj, logical block 6  [24934098.161924] Buffer I/O error on device sdaj, logical block 7 |
| --- |

则磁盘有问题，产生了坏道

使用smartctl工具检测磁盘，确认磁盘问题

| # yum install smartmontools  # smartctl -H /dev/$DRIVE |
| --- |

查看损坏磁盘的序列号Serial Number

| $ sudo smartctl -a /dev/sdc  smartctl 6.5 2016-01-24 r4214 [x86\_64-linux-4.4.0-121-generic] (local build)  Copyright (C) 2002-16, Bruce Allen, Christian Franke, www.smartmontools.org  === START OF INFORMATION SECTION ===  Device Model: TOSHIBA MG04ACA600E  Serial Number: 57J6KA41F6CD  LU WWN Device Id: 5 000039 7cb9822be  Firmware Version: FS1KUser  Capacity: 6,001,175,126,016 bytes [6.00 TB] |
| --- |

此处Serial Number: 57J6KA41F6CD

4.停止损坏的硬盘osd，并从集群从删除osd，删除osd参考文档的’删除osd’相关内容

5.点亮硬盘指示灯，方便拔出硬盘

下载并安装sas3ircu

查看所有磁盘信息

| $ sudo ./sas3ircu 0 display......Device is a Hard disk  Enclosure # : 2  Slot # : 0  SAS Address : 5003048-0-1867-f140  State : Ready (RDY)  Size (in MB)/(in sectors) : 5723166/11721045167  Manufacturer : ATA  Model Number : TOSHIBA MG04ACA6  Firmware Revision : FS1K  Serial No : 57J6KA41F6CD  Unit Serial No(VPD) : 57J6KA41F6CD  GUID : 50000397cb9822be  Protocol : SATA  Drive Type : SATA\_HDD  ...... |
| --- |

找到Serial Number: 57J6KA41F6CD对应硬盘的槽位

| Enclosure # : 2  Slot # : 0 |
| --- |

点亮硬盘指示灯

| sudo ./sas3ircu 0 locate 2:0 on |
| --- |

第一个0代表RAID卡编号，如果有多块RAID卡，那么从0开始编号，以此类推；如果只有一块RAID卡，那么就是0。

2代表Enclosure

2后面那个0代表槽位号Slot

6.关闭点亮指示灯

拔出硬盘后关闭指示灯

| sudo ./sas3ircu 0 locate 2:0 off |
| --- |

# pg故障

## pg inconsistent

集群出现pg状态变为 inconsistent 状态，inconsistent状态是pg通过scrub检测到某个或者某些对象在副本之间出现了不一致

解决方法如下

1.找到故障的pg

| ceph health detail |
| --- |

2.使用命令修复

| ceph pg repair pgid |
| --- |

## pg incomplete

在ceph peering进行数据恢复的时候，三副本的数据要通过pg\_log日志合并成一份权威的副本，当三副本的pg\_log的日志记录，不足以合并成一份权威日志的时候。Ceph集群报pg incomplete的异常。这时候要联系技术支持手动修复数据，减低数据损失。

修复过程如下

1.查找历史pg所在osd和其他副本pg所在osd，查看pg中的对象，先对比pg的主副本之间 pg里面的对象数 哪个对象数多 就把哪个pg export出来，然后import到对象数少的pg里面，选择完整对象的pg导出，再导入到主pg中（导入前备份）

然后再mark complete

| 查看对象  ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-69 --pgid 2.218 --op list  导出  ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-69/ --pgid 2.218 --op export --file 2.218  导入  ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-179/ --pgid 2.218 --op import --file 2.218 |
| --- |

修复完的pg进行cpmplete标记

| ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-179/ --pgid 2.218 --op mark-complete |
| --- |

## pg stale

pg无法及时更新自己的状态到monitor，通常原因是在peering(做元数据恢复)的时候主副本osd down了。

1. 查看stale pg所在的osd

| # ceph health detail  和  ceph pg <id> query |
| --- |

尝试启动stale pg相关的osd

| systemctl start ceph-osd@<osd\_id> |
| --- |

## pg remapped

pg remapped+peering/recovering/backfilling

pg在osd新增或删除过程中重新分布了在osd上的位置，现在pg数据正在往map后的osd上迁移。

pg active+remapped

pg的实际位置跟crush分配的位置不一致导致的。需要仔细检查osd设备树和crushmap中的对应关系，包括weight和reweight。修改crush map可恢复。

## pg down

| HEALTH\_ERR 7 pgs degraded; 12 pgs down; 12 pgs peering; 1 pgs recovering; 6 pgs stuck unclean; 114/3300 degraded (3.455%); 1/3 in osds are down  ...  pg 0.5 is down+peering  pg 1.4 is down+peering  ...  osd.1 is down since epoch 69, last address 192.168.106.220:6801/8651 |
| --- |

一般是由于故障osd，导致peering的过程卡住了。

使用 ceph pg <id> query 可以查看peering过程卡住的原因

| # ceph pg 0.5 query  { "state": "down+peering",  ...  "recovery\_state": [  { "name": "Started\/Primary\/Peering\/GetInfo",  "enter\_time": "2012-03-06 14:40:16.169679",  "requested\_info\_from": []},  { "name": "Started\/Primary\/Peering",  "enter\_time": "2012-03-06 14:40:16.169659",  "probing\_osds": [  0,  1],  "blocked": "peering is blocked due to down osds",  "down\_osds\_we\_would\_probe": [  1],  "peering\_blocked\_by": [  { "osd": 1,  "current\_lost\_at": 0,  "comment": "starting or marking this osd lost may let us proceed"}]},  { "name": "Started",  "enter\_time": "2012-03-06 14:40:16.169513"}  ]  } |
| --- |

此处peering卡住的原因为”peering is blocked due to down osds” 因为osd down 导致。启动down的osd可回复。

| systemctl start ceph-osd@<osd\_id> |
| --- |

## pg degraded+peering/recovering

pg由三副本变成二副本或者一副本，此时正在数据恢复，数据恢复完pg会处于正常状态（active+clean）

# object故障

ceph unfound object

| HEALTH\_WARN 1 pgs degraded; 78/3778 unfound (2.065%) |
| --- |

故障产生的原因：当ceph集群在recover的时候，确定了一个权威副本去修复其他副本的数据，此时去权威副本拉取数据的时候，权威副本所在的osd down，导致recover中断，object unfound。

故障处理如下

1. 找到故障objects所对应的pg

| # ceph health detail  HEALTH\_WARN 1 pgs recovering; 1 pgs stuck unclean; recovery 5/937611 objects degraded (0.001%); 1/312537 unfound (0.000%)  pg 3.8a5 is stuck unclean for 803946.712780, current state active+recovering, last acting [320,248,0]  pg 3.8a5 is active+recovering, acting [320,248,0], 1 unfound  recovery 5/937611 objects degraded (0.001%); \*\*1/312537 unfound (0.000%)\*\* |
| --- |

2.列出损坏pg的相关信息

| # ceph pg 3.8a5 query  { "state": "active+recovering",  "epoch": 10741,  "up": [  320,  248,  0],  "acting": [  320,  248,  0],  <snip>  "recovery\_state": [  { "name": "Started\/Primary\/Active",  "enter\_time": "2015-01-28 19:30:12.058136",  "might\_have\_unfound": [  { "osd": "0",  "status": "already probed"},  { "osd": "248",  "status": "already probed"},  { "osd": "301",  "status": "already probed"},  { "osd": "362",  "status": "already probed"},  { "osd": "395",  "status": "already probed"},  { "osd": "429",  "status": "osd is down"}],  "recovery\_progress": { "backfill\_targets": [], |
| --- |

查看might\_have\_unfound 这个节

already probed表面在osd中没有定位到对象

osd is down 表面osd down了，没法访问里面的数据

2.启动上一步中down的osd，保证权威副本的数据在线

| systemctl start ceph-osd@<osd\_id> |
| --- |

3.如果权威副本osd没法启动，只有放弃丢失的objects，放弃objects之前请联系技术支持工程师查看权威副本osd有没有修复的可能。如果实在不能修复权威副本osd，记录下这个object的oid (ceph pg <pgid> list\_missing )。通过object id找到对应的虚拟机，很有可能虚拟机无法重启，需要备份虚拟机的数据。

执行如下命令回滚或删除对象

| ceph pg <pgid> mark\_unfound\_lost revert|delete |
| --- |

noheartbeat\_check: no reply from

| 2016-03-21 17:43:42.426051 7f716e0a7700 -1 osd.0 70 heartbeat\_check: no reply from osd.5 since back 2016-03-21 17:43:22.233321 front 2016-03-21 17:43:41.335367 (cutoff 2016-03-21 17:43:22.426050)  2016-03-21 17:43:42.426061 7f716e0a7700 -1 osd.0 70 heartbeat\_check: no reply from osd.6 since back 2016-03-21 17:43:22.233321 front 2016-03-21 17:43:41.335367 (cutoff 2016-03-21 17:43:22.426050)  2016-03-21 17:43:42.426072 7f716e0a7700 -1 osd.0 70 heartbeat\_check: no reply from osd.7 since back 2016-03-21 17:43:22.233321 front 2016-03-21 17:43:41.335367 (cutoff 2016-03-21 17:43:22.426050)  2016-03-21 17:43:42.426075 7f716e0a7700 -1 osd.0 70 heartbeat\_check: no reply from osd.8 since back 2016-03-21 17:43:22.233321 front 2016-03-21 17:43:41.335367 (cutoff 2016-03-21 17:43:22.426050) |
| --- |

检查网络，大部分原因由于网络不稳定造成

# slow requests

| HEALTH\_WARN 30 requests are blocked > 32 sec; 3 osds have slow requests  30 ops are blocked > 268435 sec  1 ops are blocked > 268435 sec on osd.11  1 ops are blocked > 268435 sec on osd.18  28 ops are blocked > 268435 sec on osd.39  3 osds have slow requests |
| --- |

1.slow requests出现在固定3个osd上

用smartmontools 工具检查3个osd对应的硬盘是否有坏道，把有坏道或者损坏的硬盘对应osd删除。

1. slow requests 随机在集群中出现

集中数据使用率大，scrub耗时长，scrubbing过程中block了部分io。需要对scrubbing的参数调整，使scrubbing的颗粒度变小

配置如下

设置scrub的扫描时间

osd\_scrub\_begin\_hour = 0 # scrub操作的起始时间为0点

osd\_scrub\_end\_hour = 5 # scrub操作的结束时间为5点

设置scrub的扫描间隔

osd\_scrub\_sleep = 3 (参考值，根据集群实际情况调整)

每批次扫描之间的时间间隔，可以用来控制整体的扫描速度。

设置scrub扫描的workload

osd\_scrub\_chunk\_min = 1 (参考值，根据集群实际情况调整)

osd\_scrub\_chunk\_max = 1 (参考值，根据集群实际情况调整)

一次扫描的对象个数，一次性扫描1~3个对象，根据对象大小来定