# VSAN管理

## 创建

### 现状

1、创建vsan时需要选择集群；

2、为集群内的主机选择硬盘作为osd设备；

3、mon自动选择策略为：取osd个数较多的主机作为mon。

### 需求

数据中心：

创建vsan时提供两种选项：**快速创建**和**高级创建**，即在“添加数据存储”、“设备选择”中添加以下选项：

快速配置，使用现有的配置流程，增加显示快速配置下最终选择使用的主机及存储域成员、监视器的位置信息； （注:存储域成员即为osd设备）

|  |  |
| --- | --- |
| 主机名 | 存储域成员 |
| 100.1.8.5 | wwid1 |
| 100.1.8.6 | wwid2 |
| 100.1.8.7 | wwid3 |

|  |  |
| --- | --- |
| 监视器名称 | 监视器地址 |
| mon1 | 100.1.8.5 |
| mon2 | 100.1.8.6 |
| mon3 | 100.1.8.7 |

表-1. 快速配置表项

高级配置，提供以下可选项：主机列表、硬盘列表、副本数量、最小复本数、归置组数量、实际归置组数量、日志存储位置、日志大小、监视器主机列表、监视器地址、硬盘满阈值、磁盘预满阈值、最大存储空间:

|  |  |  |
| --- | --- | --- |
| **对象存储配置** | | |
| journal location | 日志存储位置 | 提供默认值 |
| journal size | 日志大小 | 提供默认值 |
| host name | 主机列表 | 提供列表选择 |
| disk list | 对象存储列表 | 提供列表选择 |
| file type | 文件系统类型 | 提供列表选择 |
| image format | 镜像类型 | 提供列表 |
| **监视器配置** | | |
| host name | 主机列表 | 提供列表选择 |
| mon addr | 监视器ip | 提供数据网端口组列表 |
| full ratio | 磁盘满阈值 | 提供默认值 |
| nearfull ratio | 磁盘预满阈值 | 提供默认值 |
| **存储池配置** | | |
| pg num | 归置组数 | 提供默认值 |
| pgp num | 有效归置组数 | 提供默认值 |
| size | 副本数 | 提供默认值 |
| min size | 最小副本数 | 提供默认值 |
| ruleset | 对象映射规则 | 提供列表选择 |
| target\_max\_bytes | 最大存储空间 | 提供默认值 |

表-2. 高级配置表项

### 快速配置

1. 首先查看vsan所在集群的所有可用主机列表；
2. 查看每个主机可用作osd的磁盘列表，在各主机的磁盘列表中选取一块作为osd，保证最终选择的osd列表容量分布具备最小的标准差，目的是保证副本的均衡和存储集群的可靠利用；（可优化：osd与cpu绑定）
3. mon参数优化：

主要通过以下步骤考虑主机的选择(TODO：待讨论)

1. 首先，划定选取范围：若osd主机列表不小于3，则从osd主机列表中按照原则b和c选取mon主机；否则，优先选取有限的osd主机，剩余按照原则b和c选取；
2. 最少主机原则：从osd主机列表中选择具有越多osd的主机，其优先级越高，目的减少不必要的主机消耗：osd\_score(host)= osd\_num(host)
3. ram优先原则：因为mon耗费ram资源较多，根据ram使用情况为各主机打分，公式为：ram\_score(host) = 1 - [N(TB) \*GB + 1GB + vm\_consume] / Total\_ram // 物理机内存，cgroup
4. 根据b和c的定义，按照主机分数从大到小选择：

主机分数：

score(host) = ram\_score(host) / osd\_num(host)

### 高级配置

功能需求：

1. 添加高级选项：默认不展开
2. 展开提示：高级模式下请咨询研发
3. 提供如表-2所示的可配置项，需按照**对象存储配置、监视器配置、存储池配置**顺序配置：
4. 所有可配置项中均提供default参数
5. pg与osd的关系

|  |  |  |
| --- | --- | --- |
| osd num | pg num | pgp num |
| (0，5) | 128 | 128 |
| [5，10) | 512 | 512 |
| [10，50) | 4096 | 4096 |

if osd\_num > 50

(OSDs \* 100)

Total PGs = ------------------ 结果取接近2的次幂

Replicas

1. 日志大小

osd journal size = {2 \* (expected throughput \* filestore max sync interval)} 单位MB

1. 参数一览表：未列出的参数均为**可选列表提供**

|  |  |  |
| --- | --- | --- |
| 参数 | 意义 | default |
| pg-num | 归置组数 | 由osd个数决定 |
| pgp-num | 使用归置组数 | 由osd个数决定 |
| size | 副本数 | 3 |
| min\_size | 处理IO的最小副数 | 2 |
| journal location | 日志存储位置 | /var/lib/ceph/osd/$cluster-$id/journal |
| journal size | 日志大小 | 由公式计算(TODO) |
| full\_ratio | 磁盘满阈值 | 95% |
| target\_max\_bytes | 最大存储空间 | 1TB |
| nearfull\_ratio | 磁盘预满阈值 | 85% |

1. 条件检查：
2. pg-num和pgp-num个数需满足2的次幂关系；
3. pgp-num不大于pg-num；
4. nearfull\_ratio不大于full\_ratio；
5. min\_size不大于size，且min\_size不为0，为了避免IO错误；
6. mon\_num不小于3；
7. TODO：osd的配置要求各osd之间容量差范围不超过某个值(可以使用均方差限制，均方差的值可以从可用磁盘列表中计算获取参考值，可以取最大和最小之间的中间值)；
8. target\_max\_bytes不超过osd的总容量；
9. 主机ram需要满足osd和mon的最低需求；
10. TODO:其他限制条件待完善。

## 删除

### 需求

在“数据存储”、“对象”中选择需要管理的vsan数据存储设备，点击删除vsan设备，提示是否删除上面的数据：

1. 添加删除vsan选项：勾选需要删除的vsan名称
2. 提示是否删除vsan上所有数据，点击确认则删除。点击否，取消删除

### 实现

删除一个节点的所有的ceph数据以及安装包

[root@node1 ~]# ceph purge // 4.0代码已经实现

## 参数配置

### 需求

主要提供vsan存储集群相关的部分关键参数修改(把mon和osd的参数配置放在“监视器管理”、“对象存储管理”中不太合适，因而全部放在了vsan的参数配置中)

### 实现

在线更改配置：

前提，已经成功创建vsan存储集群，并且集群运行状态HEALTH\_OK

参考创建部分提供的参数列表，在“数据存储”、“对象”中选择需要管理的vsan数据存储设备，在vsan数据存储设备页面单击“VSAN配置管理”选项。

参数列表:

|  |  |
| --- | --- |
| **对象存储配置** | |
| journal location | 日志存储位置 |
| journal size | 日志大小/M |
| file type | 文件系统类型 |
| image format | 镜像格式 |
| **监视器配置** | |
| full ratio | 磁盘满阈值 |
| nearfull ratio | 磁盘预满阈值 |
| **存储池配置** | |
| pg num | 归置组数 |
| pgp num | 有效归置组数 |
| size | 副本数 |
| min size | 最小副本数 |
| ruleset | 对象映射规则 |
| target\_max\_bytes | 最大存储空间 |

TODO：参数待讨论实施

对象存储配置：

监视器配置：

存储池配置：

ceph osd pool set {pool-name} {key} {value}

## 监控

### 需求

cluster监控信息位于vsan存储管理部分展示：其中pool和block device作为cluster的相关对象展示，即cluter --> pool --> block device

1. 集群的信息；
2. pool的信息以及相关对象信息；

### 实施

#### **cluster**

##### 需求

ceph存储集群以及存储池

##### 实施

|  |  |  |  |
| --- | --- | --- | --- |
| 存储集群总容量/MB | 可用容量/M | 已用容量/M | 利用率 |
| 61410 | 37792 | 23617 | 38.46% |

数据获取:

[root[@admin](http://my.oschina.net/client) ~]# ceph df

GLOBAL:

    SIZE       AVAIL       RAW USED     %RAW USED

61410M    37792M     23617M     38.46

##### 待讨论

信息放在何处展示，应该和vsan的状态信息放在一起

#### pool

##### 需求

ceph存储集群中pool的监控信息分为三部分，可以作为cluster的相关对象展示。

##### 实施

###### 容量

数据获取:

[root[@admin](http://my.oschina.net/client) ~]# ceph df

POOLS:

    NAME     ID     USED       %USED     OBJECTS

    data         0      10000M     16.28     2500

    metadata    1      3354k      0         20

    rbd          2      0          0         0

或者：

[root[@admin](http://my.oschina.net/client) ~]# ceph osd dump

epoch 88

fsid be1756f2-54f7-4d8f-8790-820c82721f17

created 2014-06-26 18:41:57.687442

modified 2014-06-30 00:46:27.179793

flags

pool 0 'data' replicated size 2 min\_size 1 crush\_ruleset 0 object\_hash rjenkins pg\_num 64 pgp\_num 64 last\_change 1 owner 0 flags hashpspool crash\_replay\_interval 45 stripe\_width 0

pool 1 'metadata' replicated size 2 min\_size 1 crush\_ruleset 0 object\_hash rjenkins pg\_num 64 pgp\_num 64 last\_change 1 owner 0 flags hashpspool stripe\_width 0

pool 2 'rbd' replicated size 2 min\_size 1 crush\_ruleset 0 object\_hash rjenkins pg\_num 64 pgp\_num 64 last\_change 1 owner 0 flags hashpspool stripe\_width 0

pool 3 'jiayuan' replicated size 2 min\_size 1 crush\_ruleset 0 object\_hash rjenkins pg\_num 256 pgp\_num 256 last\_change 73 owner 0 flags hashpspool stripe\_width 0

max\_osd 3

osd.0 up   in  weight 1 up\_from 65 up\_thru 75 down\_at 64 last\_clean\_interval [53,55) 10.240.240.211:6800/3089 10.240.240.211:6801/3089 10.240.240.211:6802/3089 10.240.240.211:6803/3089 exists,up 8a24ad16-a483-4bac-a56a-6ed44ab74ff0

osd.1 up   in  weight 1 up\_from 59 up\_thru 74 down\_at 58 last\_clean\_interval [31,55) 10.240.240.212:6800/2696 10.240.240.212:6801/2696 10.240.240.212:6802/2696 10.240.240.212:6803/2696 exists,up 8619c083-0273-4203-ba57-4b1dabb89339

osd.2 up   in  weight 1 up\_from 62 up\_thru 74 down\_at 61 last\_clean\_interval [39,55) 10.240.240.213:6800/2662 10.240.240.213:6801/2662 10.240.240.213:6802/2662 10.240.240.213:6803/2662 exists,up f8107c04-35d7-4fb8-8c82-09eb885f0e58

###### 属性

数据获取：

[root[@admin](http://my.oschina.net/client) ~]# ceph osd pool get {pool-name} {key}

例如：

[root[@admin](http://my.oschina.net/client) ~]# ceph osd pool get data pg\_num

pg\_num: 256

展示方式：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Key | pg-num | pgp-num | size | min\_size | ruleset | target\_max\_bytes |
| Value | 128 | 128 | 3 | 3 | 0 | 1TB |

###### 块设备

[root[@admin](http://my.oschina.net/client) ~]# rbd ls images

2014-05-24 17:17:37.043659 7f14caa6e700  0 -- :/1025604 >> 10.49.101.9:6789/0 pipe(0x6c5400 sd=3 :0 s=1 pgs=0 cs=0 l=1 c=0x6c5660).fault

2182d9ac-52f4-4f5d-99a1-ab3ceacbf0b9

34e1a475-5b11-410c-b4c4-69b5f780f03c

476a9f3b-4608-4ffd-90ea-8750e804f46e

###### 详情

以name为关键字按列显示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 标识 | 已用容量 | 占用率 | 对象数目 | 最大容量 |
| data | 0 | 10000M | 16.28% | 2500 | 1TB |
| 归置组数 | 使用归置组数 | 副本数 | 最小副本数 | 映射规则 |  |
| 128 | 128 | 3 | 2 | 0 |  |

镜像列表：（提供折叠显示” + “查看详细信息）rbd image list

2182d9ac-52f4-4f5d-99a1-ab3ceacbf0b9

34e1a475-5b11-410c-b4c4-69b5f780f03c

74cb427c-cee9-47d0-b467-af217a67e60a

…..

##### 待讨论

信息放在何处展示

# 监视器管理

## 需求

前提：已经创建完成vsan存储，并且存储集群状态为HEALTH\_OK，在“数据存储”、“对象”中选择需要管理的vsan数据存储设备，在vsan数据存储设备页面单击“监视器管理”选项，提供如下功能：

1. 选项卡“在线迁移”：提供mon在线迁移功能
2. 选项卡“监视器信息”：提供mon的监控信息展示

## 实现

### 在线迁移

#### 需求

用户通过“监视器管理”进入，迁移提供“源主机列表”和“目的主机列表”，根据用户的输入进行迁移前检查，迁移，迁移后检查。

#### 实现

###### 进入mon管理

通过点击“mon管理”，进入“在线迁移”选项卡，展示当前集群的模拟位置信息。

|  |  |  |  |
| --- | --- | --- | --- |
| 集群 | 监视器 | 源主机 | 目的主机 |
| Ceph1 | mon.a | xxx | XXXX修改 |
| mon.b | xxx | XXXX修改 |
| mon.c | xxx | XXXX修改 |
| Ceph2 | mon.d | xxx | 修改 |
| mon.e | xxx | 修改 |
| mon.f | xxx | 修改 |

迁移前检查：

1. 集群状态检查

[root@admin ~]#ceph health //查看集群状态

状态为HEALTH\_OK的集群显示为可修改mon（可点击），白色背景；其他状态集群显示为不可修改mon（不可点击），灰色背景，不可修改。

1. 法定人数检查

判断mon的格式是否满足修改mon的条件（n-1）≥[n/2]。其中n是mon数量。满足，则可修改，否则显示为不可修改mon（不可点击），灰色背景，不可修改。

###### 参数检查

1. 点击“修改”，输入新的mon ip后，对输入ip进行合法性和合理性检查：即检查输入ip值是否合法，同时检查是否与集群其他mon addr在同一网段。
2. 限制：一次操作仅允许修改单个mon的ip，若出现多次修改，单击“保存”时提示请依次修改。

###### 删除mon

[root@admin ~]#service ceph -a stop mon.[mon\_id] //停掉要修改的mon

[root@admin ~]#ceph mon remove {mon\_id} //删除mon

###### 添加mon

[root@admin ~]#ssh {new\_mon\_host}//登录到新的mon上，在这之前需要拷贝密钥到新的mon

[root@admin ~]#sudo mkdir /var/lib/ceph/mon/ceph-{mon-id} //在新的mon上创建默认目录

[root@admin ~]#cd /var/lib/ceph/mon/ceph-{mon-id} //进入目录

[root@admin ~]#touch sysvinit done

[root@admin ~]#exit //退出新的mon

[root@admin ~]#mkdir {tmp} //创建新的tmp文件夹用来保存中间文件

[root@admin ~]#ceph auth get mon. -o {tmp}/{filename\_1}//备份原集群的keyring到{tmp}下

[root@admin ~]#ceph mon getmap -o {tmp}/{filename\_2} //备份原集群的monmap到{tmp}下

[root@admin ~]#sudo ceph-mon -i {mon-id} --mkfs --monmap {tmp}/{filename} —keyring {tmp}/{filename}

[root@admin ~]#monmaptool --rm mon.{mon\_id} //删除要删除的mon信息

[root@admin ~]#ceph mon add <mon-id> <ip>[:<port>] //添加新的mon

[root@admin ~]#ceph-mon -i {mon-id} --public-addr {ip:port}

###### 修改ceph.conf

从配置文件中删除原有mon的信息，添加迁移后的mon信息。

###### 启动新的mon

[root@admin ~]#service ceph -a start mon.[mon\_id] //启动新加的mon

### 监控

#### 需求

界面上展示mon节点，mon服务运行状态

#### 实现

|  |  |  |  |
| --- | --- | --- | --- |
| name | addr | rank | quorum |
| node1/ mon.a | 10.240.240.211:6789/0 | 0 (leader) | y |
| node2/ mon.b | 10.240.240.212:6789/0 | 1 | y |
| node3/ mon.c | 10.240.240.213:6789/0 | 2 | y |

1、ceph mon stat（推荐）

e2: 3 mons at {node1=10.240.240.211:6789/0,node2=10.240.240.212:6789/0,node3=

10.240.240.213:6789/0}, election epoch 294, quorum 0,1,2 node1,node2,node3

第一位是选举的leader

2、mon\_map

获取mon\_map: ceph mon getmap -o {tmp}/{filename}

输 出：monmaptool --print {tmp}/{filename}

monmaptool: monmap file {tmp}/{filename}

epoch 1

fsid 224e376d-c5fe-4504-96bb-ea6332a19e61

last\_changed 2012-12-17 02:46:41.591248

created 2012-12-17 02:46:41.591248

0: 10.0.0.1:6789/0 mon.a

1: 10.0.0.2:6789/0 mon.b

2: 10.0.0.3:6789/0 mon.c

3、查看mon的dump信息

[root@admin~]# ceph mon dump

dumped monmap epoch 2

epoch 2

fsid be1756f2-54f7-4d8f-8790-820c82721f17

last\_changed 2014-06-26 18:43:51.671106

created 0.000000

0: 10.240.240.211:6789/0 mon.node1

1: 10.240.240.212:6789/0 mon.node2

2: 10.240.240.213:6789/0 mon.node3

# 对象存储管理

## 需求

前提：已经创建完成vsan存储，并且存储集群状态为HEALTH\_OK，在“数据存储”、“对象”中选择需要管理的vsan数据存储设备，在vsan数据存储设备页面单击“对象存储管理”选项，提供如下功能：

1. 选项卡“扩容”：选择一个不是OSD的主机及其本地存储池（支持xattr），部署为OSD
2. 选项卡“减容”：移除 OSD
3. 监控信息完善

## 实现

### 扩容

#### 需求

已经创建完成vsan存储，并且存储集群状态为HEALTH\_OK，在“数据存储”、“对象”中选择需要管理的vsan数据存储设备，在vsan数据存储设备页面单击“对象存储管理”选项，选择“扩容”选项卡：提供将一个不是OSD的主机，将其本地存储池（支持xattr），部署为OSD，提供列表：

|  |  |  |
| --- | --- | --- |
| 主机列表 | 磁盘列表 | 参数设置 |

其中：

主机列表仅提供当前未部署过osd的主机；

磁盘列表支持同时选择多块磁盘；

参数设置支持：journal location、journal size、file type

#### 实施

1. 磁盘容量检查

新加入的osd与存储域中osd成员容量不能超过某个门限值，若超过则告警，不建议将该块盘作为osd部署；

1. 添加osd

逐个将选择硬盘部署为osd设备

1. ceph osd create [{uuid}]
2. ssh {new-osd-host}

sudo mkdir /var/lib/ceph/osd/ceph-{osd-number}

1. sudo mkfs -t {fstype} /dev/{drive}

sudo mount -o user\_xattr /dev/{hdd} /var/lib/ceph/osd/ceph-{osd-number}

1. ceph-osd -i {osd-num} --mkfs –mkkey
2. ceph auth add osd.{osd-num} osd 'allow \*' mon 'allow rwx' -i /var/lib/ceph/osd/ceph-{osd-num}/keyring
3. ceph osd crush set {id-or-name} {weight} root={pool-name} [{bucket-type}={bucket-name} ...]
4. sudo /etc/init.d/ceph start osd.{osd-num}
5. 检查该osd是否生效

[root@admin~]# ceph osd stat

      osdmap e88: 3 osds: 3 up, 3 in

判断up和in的个数是否与添加后的一致；

### 减容

#### 需求：

已经创建完成vsan存储，并且存储集群状态为HEALTH\_OK，在“数据存储”、“对象”中选择需要管理的vsan数据存储设备，在vsan数据存储设备页面单击“对象存储管理”选项，选择“减容”选项卡：提供主机列表和存储域成员即osd，由用户勾选需要删除的硬盘；

#### 实施

进入“减容”选项卡，提供如下列表：

|  |  |
| --- | --- |
| 主机列表 | 存储域成员列表 |

减容需求4.0中已经实现；

不支持单次操作删除多个存储域成员；

参考：减容的步骤

1. 检查

删除存储域成员后存储域的容量是否达到full\_ratio或者nearfull\_ratio，若达到，不允许减容操作。

1. 移出集群

ceph osd out {osd-num}

1. 停止osd

ssh {osd-host}

/etc/init.d/ceph stop osd.{osd-num}

1. 移出CRUSH map

ceph osd crush remove {name} // osd.1

1. 删除key

ceph auth del osd.{osd-num}

1. 删除osd

ceph osd rm {osd-num}

1. 查找ceph.conf文件，删除osd-num匹配项
2. 拷贝配置文件到各个host

### 监控

#### 需求

已经创建完成vsan存储，在“数据存储”、“对象”中选择需要管理的vsan数据存储设备，在vsan数据存储设备页面单击“对象存储管理”选项，界面上展示OSD节点，OSD服务运行状态。

#### 实施

展示界面：

|  |  |  |  |
| --- | --- | --- | --- |
| osd\_name | up/down(数据服务状态) | in/out(集群服务状态) | weight |
| osd.0 | up | in | 1 |
| osd.1 | up | in | 1 |
| osd.2 | up | in | 1 |

1、查看osd状态

[root@admin~]# ceph osd stat

      osdmap e88: 3 osds: 3 up, 3 in

2、查看osd的映射信息

[root[@admin](http://my.oschina.net/client) ~]# ceph osd dump

epoch 88

fsid be1756f2-54f7-4d8f-8790-820c82721f17

created 2014-06-26 18:41:57.687442

modified 2014-06-30 00:46:27.179793

flags

pool 0 'data' replicated size 2 min\_size 1 crush\_ruleset 0 object\_hash rjenkins pg\_num 64 pgp\_num 64 last\_change 1 owner 0 flags hashpspool crash\_replay\_interval 45 stripe\_width 0

pool 1 'metadata' replicated size 2 min\_size 1 crush\_ruleset 0 object\_hash rjenkins pg\_num 64 pgp\_num 64 last\_change 1 owner 0 flags hashpspool stripe\_width 0

pool 2 'rbd' replicated size 2 min\_size 1 crush\_ruleset 0 object\_hash rjenkins pg\_num 64 pgp\_num 64 last\_change 1 owner 0 flags hashpspool stripe\_width 0

pool 3 'jiayuan' replicated size 2 min\_size 1 crush\_ruleset 0 object\_hash rjenkins pg\_num 256 pgp\_num 256 last\_change 73 owner 0 flags hashpspool stripe\_width 0

max\_osd 3

osd.0 up   in  weight 1 up\_from 65 up\_thru 75 down\_at 64 last\_clean\_interval [53,55) 10.240.240.211:6800/3089 10.240.240.211:6801/3089 10.240.240.211:6802/3089 10.240.240.211:6803/3089 exists,up 8a24ad16-a483-4bac-a56a-6ed44ab74ff0

osd.1 up   in  weight 1 up\_from 59 up\_thru 74 down\_at 58 last\_clean\_interval [31,55) 10.240.240.212:6800/2696 10.240.240.212:6801/2696 10.240.240.212:6802/2696 10.240.240.212:6803/2696 exists,up 8619c083-0273-4203-ba57-4b1dabb89339

osd.2 up   in  weight 1 up\_from 62 up\_thru 74 down\_at 61 last\_clean\_interval [39,55) 10.240.240.213:6800/2662 10.240.240.213:6801/2662 10.240.240.213:6802/2662 10.240.240.213:6803/2662 exists,up f8107c04-35d7-4fb8-8c82-09eb885f0e58

# 服务管理

## 现状

未提供接口封装，代码未实现

## 需求

封装服务起停管理的接口，包括vsan服务、osd服务、mon服务、指定的osd服务、指定的mon服务的接口。

## 实现

### 基本格式

推荐使用sysvint管理起停服务

命令参考：{commandline} [options] [commands] [daemons]

/etc/init.d/ceph [options] [start|restart] [daemonType|daemonID]

| **options** | **Shortcut** | **Description** |
| --- | --- | --- |
| --verbose | -v | Use verbose logging. |
| --valgrind | N/A | (Dev and QA only) Use [Valgrind](http://www.valgrind.org/) debugging. |
| --allhosts | -a | Execute on all nodes in ceph.conf. |
| --restart | N/A | Automatically restart daemon if it core dumps. |
| --norestart | N/A | Don’t restart a daemon if it core dumps. |
| --conf | -c | Use an alternate configuration file. |
| **command** | **Description** | |
| start | Start the daemon(s). | |
| stop | Stop the daemon(s). | |
| forcestop | Force the daemon(s) to stop. Same as kill -9 | |
| killall | Kill all daemons of a particular type. | |
| cleanlogs | Cleans out the log directory. | |
| cleanalllogs | Cleans out everything in the log directory. | |

### 存储集群服务

语法： /etc/init.d/ceph start/stop 在本节点上执行

/etc/init.d/ceph -a start/stop -a在所有节点上执行

### 进程类型服务

语法：/etc/init.d/ceph start/stop {daemon-type} 在本节点上执行

/etc/init.d/ceph -a start/stop {daemon-type} -a在所有节点上执行

例如：

sudo /etc/init.d/ceph start/stop osd

sudo /etc/init.d/ceph -a start/stop osd

### 指定进程服务

语法：sudo /etc/init.d/ceph start {daemon-type}.{instance} 在本节点上执行

sudo /etc/init.d/ceph start -a {daemon-type}.{instance} -a在所有节点上执行

例如：

sudo /etc/init.d/ceph -a start/stop osd.0

# 在线维护

## 需求

支持在线更换硬盘（该盘在存储集中作为osd使用）

进入在线维护模式，选择“更换硬盘”，手动热拔插更换硬盘。

提供如下所示列表：

|  |  |
| --- | --- |
| 主机列表 | 存储域成员列表 |

选择需要更换硬盘的主机和存储域成员；

限制：单次操作仅支持更换一块硬盘；

## 实现

### 检查

预测将硬盘移除后存储池的容量使用是否超过full\_ratio，若超过则提示“容量利用率过高，请先扩容后再更换硬盘”，禁止更换硬盘。

### 停止osd迁移

[root[@admin](http://my.oschina.net/client) ~]# ceph osd set noout

### 删除osd

参考“减容”步骤；

### 更换硬盘和检查

识别新加入的硬盘（TODO：讨论如何实现）

如果第1步检查通过，那么这一步就不会存在已用容量超过full\_ratio的限制；

### 添加osd

参考“扩容”步骤，其中已经包含了对硬盘容量的检查；

### 开启数据迁移

[root[@admin](http://my.oschina.net/client) ~]# ceph osd set out

### 检查集群状态

检查更换完硬盘后集群状态是否为HEALTH\_OK

# 离线维护

## 需求

进入离线维护模式，支持跨网段修改VSAN IP

## 实施

### “进入维护模式”

通过点击“进入离线维护”按钮，进入离线维护模式：

[root@admin ~]# /etc/init.d/ceph -a stop

### 输入修改信息

进入“离线维护”，单击修改“监视器ip”选项卡：

注意：离线维护适用于数据中心迁移等非频繁操作，请谨慎使用

|  |  |  |
| --- | --- | --- |
| mon详情 | 源ip | 目标ip |
| mon.a | 10.0.0.1:6789/0 | 10.1.0.1:6789/0 |
| mon.b | 10.0.0.2:6789/0 | 10.1.0.2:6789/0 |
| mon.c | 10.0.0.3:6789/0 | 10.1.0.3:6789/0 |

### 参数检查

输入需要修改的ip后，单击”保存“按钮，系统检查输入修改后的ip是否属于同一网段，即ip合法性和合理性检查，检查通过进行下一步操作。否则，单击确认后弹出“参数输入有误，请检查重新输入”告警，不修改ip。

### 备份

源mon的data数据进行本分后删除。

### 修改ip

1. [root@admin ~]# ceph mon getmap -o 1.txt // 1、导出mon map到文件1.txt
2. [root@admin ~]# monmaptool --rm a --rm b --rm c 1.txt // 2、移除原有的mon
3. [root@admin ~]# monmaptool --add a 10.1.0.1:6789 --add b 10.1.0.2:6789 --add c 10.1.0.3:6789 1.txt // 3、加入修改过ip的mon
4. [root@admin ~]# monmaptool --print 1.txt // 4、检查是否修改成功
5. [root@admin ~]# /etc/init.d/ceph stop mon // 6、暂停mon服务
6. [root@admin ~]# ceph-mon -i {mon-id} --inject-monmap 1.txt //7、注入新的mon map

### 修改ceph.conf

查找ceph.conf中是否存在原有ip，若存在，则替换为修改后的ip

### 退出离线维护

ip修改成功以后，默认自动退出离线维护模式

[root@admin ~]# ceph osd unpause

TODO: 如何添加退出离线维护模式

# 块设备管理

## 前提

已经开启并运行vsan服务；

ceph使用rbd命令来使用RADOS块设备镜像，即RBD镜像。RBD镜像是一种简单的块设备，被条带化为对象存储于RADOS对象存储。限制：RBD镜像被划分为条带化对象的个数需要满足2的次幂关系。

现状：由上层逻辑直接调用rbd命令进行使用和管理。

## 需求

封装接口，提供给上层逻辑调用。

## 操作

### 创建

1. 首先创建pool资源池，用来存储信息

[root@admin ~]# ceph osd pool create {pool-name} {pg-num} [{pgp-num}] [replicated]

1. 为块设备创建镜像（即RBD镜像）

注意：如果不指定--pool {pool-name}，默认为rbd pool

[root@admin ~]# rbd create {image-name} --size {megabytes} --pool {pool-name}

或者

[root@admin ~]# rbd create {image-name}/{pool-name} --size {megabytes}

附:

--size 单位为/M

--object 对象大小，按照次幂定义：如4M为22

--format 指定输出格式：plain，json，xml

### 映射

### 查询

查询存储池中的块设备信息

[root@admin ~]# rbd ls // 默认list出rbd pool中的block devices

[root@admin ~]# rbd ls {poolname} // list指定pool中的block devices

查询存储池中的镜像信息

[root@admin ~]# rbd --image {image-name} -p {pool-name} info

例如：

[root@admin ~]# rbd --image rbdpoolimages -p rbdpool info

rbd image 'rbdpoolimages': // image的名称

       size 80960 MB in 20240 objects // image的大小，和其中包含对象个数

       order 22 (4096 kB objects) // 其中每个对象的大小为4M

       block\_name\_prefix: rb.0.1272.238e1f29

       format: 1 // 对象分布的格式

format：1 支持所有版本的librbd和内核rbd模块，但是不支持某些新功能，比如克隆。

format：2 仅支持librbd，不支持内核rbd模块，对新功能支持较好。

### 调整大小

ceph块设备镜像是thin类型的，通过在创建时指定了--size其大小。

使用下列操作调整大小：

[root@admin ~]# rbd resize --image {image-name} --size {megabytes}

### 移除

[root@admin ~]# rbd rm {image-name} -p {pool-name} # 不指定pool-name,默认为rbd

### 快照

#### 前提

对镜像做快照前要停止镜像的io

#### 创建

创建指定镜像的快照副本：

[root@admin ~]# rbd --pool {pool-name} snap create --snap {snap-name} {image-name}

或者

[root@admin ~]# rbd snap create {pool-name}/{image-name}@{snap-name}

例如：

[root@admin ~]# rbd --pool rbd snap create --snap snapname foo

[root@admin ~]# rbd snap create rbd/foo@snapname

#### 查看

查看指定镜像的快照副本信息：

[root@admin ~]# rbd --pool {pool-name} snap ls {image-name}

或者

[root@admin ~]# rbd snap ls {pool-name}/{image-name}

#### 回滚

用snap-name快照副本数据覆写当前磁盘镜像内容：

[root@admin ~]# rbd --pool {pool-name} snap rollback --snap {snap-name} {image-name}

或者

[root@admin ~]# rbd snap rollback {pool-name}/{image-name}@{snap-name}

#### 删除

删除指定镜像的指定快照副本：

[root@admin ~]# rbd --pool {pool-name} snap rm --snap {snap-name} {image-name}

或者

[root@admin ~]# rbd snap rm {pool-name}/{image-name}@{snap-name}

#### 清洗

删除指定镜像的所有快照副本：

[root@admin ~]# rbd --pool {pool-name} snap purge {image-name}

或者

[root@admin ~]# rbd snap purge {pool-name}/{image-name}

#### 分层

ceph支持为块设备的只读快照副本创建COW克隆。一般步骤：为块设备镜像创建快照副本，开启保护快照副本，为快照副本创建 多个COW克隆；克隆的创建是快速的，因为快照副本只读，克隆仅仅需要保存快照副本的索引，该索引能够保证对快照副本的快速读，以此快照副本为基础，克隆仅保存写入信息。

该COW克隆具有所有ceph块设备镜像的操作属性。

##### 保护快照

创建COW 克隆前，需要先保护快照，避免源快照的丢失及损坏：

[root@admin ~]# rbd --pool {pool-name} snap protect --image {image-name} --snap {snapshot-name}

或者

[root@admin ~]# rbd snap protect {pool-name}/{image-name}@{snapshot-name}

##### 克隆快照

注意：克隆功能仅适用于format格式为2的image块设备

[root@admin ~]# rbd --pool {pool-name} --image {parent-image} --snap {snap-name} --dest-pool {pool-name} --dest {child-image}

或者

[root@admin ~]# rbd clone {pool-name}/{parent-image}@{snap-name} {pool-name}/{child-image-name}

例如：

rbd clone rbd/my-image@my-snapshot rbd/new-image

##### 取消保护

[root@admin ~]# rbd --pool {pool-name} snap unprotect --image {image-name} --snap {snapshot-name}

或者

[root@admin ~]# rbd snap unprotect {pool-name}/{image-name}@{snapshot-name}

例如：

[root@admin ~]# rbd --pool rbd snap unprotect --image my-image --snap my-snapshot

[root@admin ~]# rbd snap unprotect rbd/my-image@my-snapshot

##### 查看克隆

[root@admin ~]# rbd --pool {pool-name} children --image {image-name} --snap {snap-name}

或者：

[root@admin ~]# rbd children {pool-name}/{image-name}@{snapshot-name}

##### 解除索引

由于一个快照副本可以创建多个COW克隆镜像，克隆镜像中保存着对快照副本的索引，因此在删除该快照副本时，在解除COW克隆镜像的关联之前，首先将快照的数据全部拷贝到克隆镜像中，类似于合并差异，然后再删除索引信息；这样，克隆镜像就保存了快照的全部信息，故占用的空间将会与原来增加。

[root@admin ~]# rbd --pool {pool-name} flatten --image {image-name}

[root@admin ~]# rbd flatten {pool-name}/{image-name}

例如：

[root@admin ~]# rbd --pool rbd flatten --image my-image

[root@admin ~]# rbd flatten rbd/my-image

## 监控

### 需求

将集群中的快设备信息集中展示；

### 实施

查看ceph中的指定块设备的使用信息，提供image列表，每个image名称下以**”+”**折叠显示块设备信息。作为pool的相关对象显示：

1、查询ceph存储集群中的pool信息 // ceph osd lspools

2、查询所有pool的image list // rbd ls images

3、查询每个image 的详细信息

1. 基本信息：--format 可以设置输出格式为json

[root[@admin](http://my.oschina.net/client) ~]# rbd info -p images --image 74cb427c-cee9-47d0-b467-af217a67e60a

rbd image '74cb427c-cee9-47d0-b467-af217a67e60a':

        size 1048 MB in 131 objects // 镜像空间大小

        order 23 (8192 KB objects) // 对象大小：8M

        block\_name\_prefix: rbd\_data.95c7783fc0d0

        format: 2 // 镜像格式：1-支持克隆，2-不支持克隆

        features: layering

1. 快照

rbd snap ls [image-name]

1. 克隆

rbd children [pool-name]/[image-name]@mysnap

例如：TODO:待测试

[root[@admin](http://my.oschina.net/client) ~]# rbd children pool/image@snap

pool2/child1

pool2/child2

1. 详情

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 镜像名称 | 归属存储池 | 容量 | 镜像格式 | 相关对象 | |
| XXX | rbd | 1048 MB | 2 | 快照列表 | 克隆列表 |

注意：镜像格式1-支持克隆，2-不支持克隆

## HELP

### 主要参数

--image-format  定义对象的layout格式，默认使用1

--size  大小单位为Mb

--order  对象的大小，单位为bits

--format 设置输出格式，可以是json、plain

-o map-options, --options map-options 选择映射镜像

### 命令行

具体参考：<http://docs.ceph.com/docs/v0.80.5/man/8/rbd/>

**ls** [-l | –long] [pool-name] 列出pool中所有rbd镜像，l为附加显示快照，long附加显示大小、父本和格式等信息

**info** [image-name] 展示这个镜像的所有信息，包括父本以及是否被保护等信息

# vsan网络配置

## 需求

VSAN网络配置中仅提供可用的数据网络端口组用于VSAN网络绑定

## 实施

主机 【**管理**】 【**存储**】 【 **VSAN 网络配置**】单击【**添加**】选项，弹出内容为：

|  |  |
| --- | --- |
| 端口组名称 | IP |
| dataNetwork | XXX.XXX.XXX.XXX |

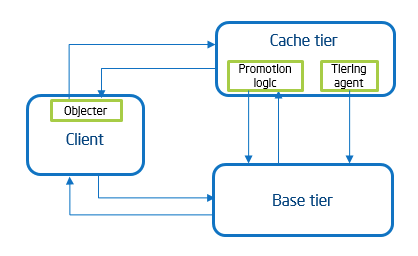
即后台过滤业务类型为“管理网端口组”显示列表，仅提供业务类型为“数据网端口组“列表。

# cache tier

## 背景介绍：

分级缓存可提升后端存储内某些（热点）数据的 I/O 性能。分级缓存需创建一个由高速而昂贵存储设备（如 SSD ）组成的存储池、作为缓存层，以及一个相对低速/廉价设备组成的后端存储池（或纠删码编码的）、作为经济存储层。

Ceph存储集群生产环境中可以采用以高性能的SSD介质作为Cache Tier实现分级缓存。



Ceph 的对象处理器决定往哪里存储对象，缓存层代理自动处理缓存层和后端存储之间的数据迁移，管理员可以指定迁移规则，通过参数配置调整实现。所以缓存层和后端存储层对 Ceph 客户端来说是完全透明的。

将一个基于SSD构建的cache pool，置于基于HDD的rbd pool之前作为缓存层，通过CRUSH规则映射SSD和cache pool，实现Ceph分级缓存。可以提供的cache模式包括：

* writeback ： Ceph 客户端会把数据写入缓存层、并收到缓存层发来的 ACK ；写入缓存层的数据会被迁移到存储层、即从缓存层下刷。当 Ceph 客户端要读取的数据位于存储层时，缓存层代理会把这些数据迁移到缓存层，然后再发往 Ceph 客户端。这样一来，**Ceph 客户端将直接与缓存层进行 I/O 操作**，直到数据不再被读写。由于缓存层上的数据时刻是最新的，故此模式对于易变数据来说较理想（如照片/视频编辑、事务数据等：写操作较多）。
* readonly ： **Ceph 客户端直接把数据写入后端**。读取时，Ceph 把相应对象从后端复制到缓存层，根据已定义策略、脏对象会被缓存层踢出。易变数据不要用 readonly 模式，可能导致数据不一致（原因：易变数据写操作多，会引起存储层数据在向缓存层复制的过程中读写操作数据不一致）。

## 需求列表

前提：原有所有VSAN接口不变

新增需求：

* 添加Cache层
* 配置Cache参数
* Cache层扩展
* Cache层缩减
* 删除Cache层

## 添加cache层

### 需求

本处的内容列出的是在规划存储集群时的配置参考：

### 创建ssd-disk OSD

// 参考创建osd流程

ceph-deploy osd create {node-name}:{ssd-disk}[:{path/to/journal}]

### 添加crush rule-set

* 添加设备

device 16 osd.16

device 20 osd.20

* 修改crush map，添加bucket

1. host host1-ssd {

id -1

alg straw

hash 0

item osd.16 weight 1.00

}

1. host host2-ssd {

id -2

alg straw

hash 0

item osd.20 weight 1.00

}

1. root ssd {

id -3

alg straw

hash 0

item host1-ssd weight 1.000

item host2-ssd weight 1.000

}

1. 添加rule-set

rule cachetier {

ruleset 4

type replicated

min\_size 0

max\_size 4

step take ssd

step chooseleaf firstn 0 type host

step emit

}

### 创建cachepool 绑定

ceph osd pool create {pool-name/cachepool } {pg-num} [{pgp-num}] [replicated] [crushruleset-name/ cachetier]

ceph osd pool set {cachepool } crush\_ruleset 4

### 将cachepool挂接后端存储池

ceph osd tier add {storagepool} {cachepool}

### 设置缓存模式

ceph osd tier cache-mode {cachepool} {cache-mode} writeback // 建议writeback

### 流量迁移

必须把所有客户端流量从存储池迁移到缓存存储池

ceph osd tier set-overlay {storagepool} {cachepool}

## 删除ceche层

### 需求

考虑到只读模式会带来数据不一致的问题，建议使用写回模式，故这里也只提供写回模式的流程参考。

回写缓存可能含有更改过的数据，所以在禁用并去除前，必须采取些手段以免丢失缓存内近期更改的对象

### 实施

#### 1、更改cache-mode为forward

把缓存模式改为 forward ，这样新的和更改过的对象将直接刷回到后端存储池

ceph osd tier cache-mode {cachepool} forward

#### 2、确保缓存池已刷回

查看pool中所有对象是否刷回：

rados -p {cachepool} ls

可以手动刷回：

rados -p {cachepool} cache-flush-evict-all

#### 3、移除覆盖

ceph osd tier remove-overlay {storagetier}

#### 4、剥离缓存层存储池

ceph osd tier remove {storagepool} {cachepool}

## 配置

### 需求

语法设置：ceph osd pool set {cachepool} {key} {value}

缓存空间：

* 刷回： 代理找出修改过（或脏）的对象、并把它们转发给存储池做长期存储。
* 赶出： 代理找出未修改（或干净）的对象、并把最近未用过的赶出缓存

### 实施

#### 1、目标尺寸和类型

ceph osd pool set {cachepool} hit\_set\_type bloom // 生产环境仅提供bloom类型过滤

ceph osd pool set {cachepool} hit\_set\_count 1 // 保留的集中个数

ceph osd pool set {cachepool} hit\_set\_period 3600 // 统计时间区间

ceph osd pool set {cachepool} target\_max\_bytes 1000000000000

#### 2、Cache相对大小，百分比

cache\_target\_dirty\_ratio: 脏数据占多少时，向Backing Tier写回数据

cache\_target\_dirty\_high\_ratio: 脏数据占多少时，快速向Backing Tier写回数据

cache\_target\_full\_ratio: 使用多少容量时淘汰数据，保证容量

#### 3、Cache绝对大小，数值

提供按照数据量和对象数目另种度量标准

target\_max\_bytes {#bytes}: 指定最大的数据量，到线写回

target\_max\_objects {#objects}: 指定最大的对象数目

#### 4、Cache Age

规定缓存层代理必须延迟多久才能把脏对象刷回后端存储池

cache\_min\_flush\_age {#seconds}: 脏数据写回计时

cache\_min\_evict\_age {#seconds}: 净数据淘汰计时

## 扩容

### 需求

cache tier的扩展本质是添加osd到指定的pool中。

### 实施

需要停止数据迁移

### 停止数据迁移

避免新加入的ssd osd被均衡，造成后续重复迁入迁出数据消耗ram

[root[@admin](http://my.oschina.net/client) ~]# ceph osd set noout

### 创建ssd osd

步骤参考添加存储对象管理“扩容”

区别：需要修改CRUSH map，将ssd osd位于指定rule\_set中；

### 开启数据迁移

[root[@admin](http://my.oschina.net/client) ~]# ceph osd set out

## 减容

### 需求

cache层缩减的本质就是将osd移除cachepool，即将相应的osd与cachepool解除绑定关系，对于移除的osd设备处理建议是采取默认处理，这样CRUSH将会将空闲osd重新均衡到data pool中，提高存储集群的空间利用率；

### 实施

#### 获取CRUSH map

ceph osd getcrushmap -o {compiled-crushmap-filename}

// 输出文件compiled-crushmap-filename

#### 反编译CRUSH map

crushtool -d {compiled-crushmap-filename} -o {decompiled-crushmap-filename}

// 反编译以后的decompiled-crushmap-filename

#### 编辑其中的Devices, Buckets和Rules，指定rule-id

1. host host1-ssd

item osd.16 weight 1.000

1. ~~host host2-ssd~~

~~item osd.20 weight 1.000~~

1. root ssd

item host1-ssd weight 1.000 ?

~~item host2-ssd weight 4.000~~

#### 重新编译CRUSH map

crushtool -c {decompiled-crush-map-filename} -o {compiled-crush-map-filename}

// 重新编译以后的文件compiled-crush-map-filename

#### 设置CRUSH map

ceph osd setcrushmap -i { compiled-crush-map-filename }

# RBD MAP

## 需求

实现rbd块设备映射管理接口封装

## 实施

以一个实例展示：

### 新建rbd

rbd create test1 --size 1024000 // 大小为1TB

### 挂载rbd

rbd map test1 --pool rbd --name client.admin // 挂载给客户端

### 查看rbd信息

root@cloud2:~# rbd info test1

rbd image 'test1':

size 1000 GB in 256000 objects

order 22 (4096 KB objects)

block\_name\_prefix: rb.0.18e4.6b8b4567

format: 1

上面的输出中block\_name\_prefix就是这个rbd所有object名的prefix，要实现快速删除rbd这个prefix相当有用。

root@wocloud2:~# rados ls -p rbd|grep '^rb.0.18e4.6b8b4567'|wc -l

160

可以看到1T左右的rbd块包含160个object

### 格式化rbd块

root@cloud2:~# time mkfs.xfs /dev/rbd/rbd/test1

log stripe unit (4194304 bytes) is too large (maximum is 256KiB)

log stripe unit adjusted to 32KiB

meta-data=/dev/rbd/rbd/test1 isize=256 agcount=33, agsize=8190976 blks

= sectsz=512 attr=2, projid32bit=0

data = bsize=4096 blocks=262144000, imaxpct=25

= sunit=1024 swidth=1024 blks

naming =version 2 bsize=4096 ascii-ci=0

log =internal log bsize=4096 blocks=128000, version=2

= sectsz=512 sunit=8 blks, lazy-count=1

realtime =none extsz=4096 blocks=0, rtextents=0

real 0m19.499s

user 0m0.004s

sys 0m0.212s

1T左右的rbd块格式化事件约在19s的时间。

### 删除rbd

删除前先取消挂载

rbd unmap /dev/rbd/rbd/test1

#### 直接删除法

直接删除

root@wocloud2:~# time rbd rm test1

Removing image: 100% complete...done.

real 0m26.850s

user 0m43.328s

sys 0m32.508s

如上可知，大致的删除时间为27s左右。

#### 快速删除法

先删除header files  
root@wocloud2:~# time rados rm -p rbd test1.rbd  
real 0m0.026s  
user 0m0.004s  
sys 0m0.016s

再删除rbd数据块  
root@wocloud2:~# time rados -p rbd ls|grep '^rb.0.1607.6b8b4567'|xargs -n 200 rados -p rbd rm  
real 0m0.748s  
user 0m0.172s  
sys 0m0.064s  
前提是获得这个prefix

最后删除rbd  
root@wocloud2:~# time rbd rm test1  
2014-08-15 15:41:42.749394 7f8b5f09d780 -1 librbd::ImageCtx: error finding header: (2) No such file or directory  
Removing image: 100% complete...done.  
real 0m0.030s  
user 0m0.004s  
sys 0m0.020s

# 遗留问题

待完善…