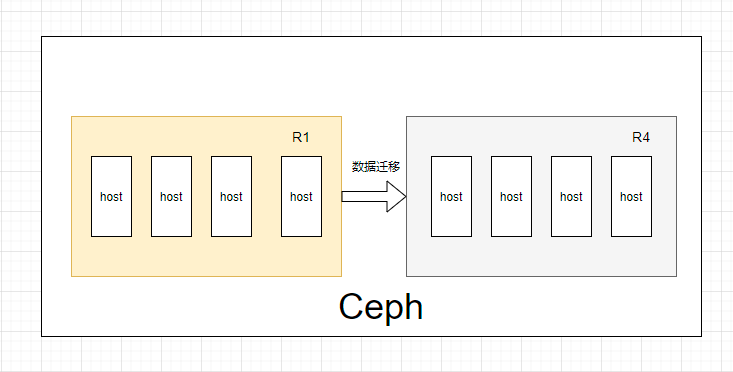
R1--R4数据迁移

# 需求

R1的数据都迁移到R4

R1的服务器下线



# osd的迁移

## 带宽要求

迁移的时候非常消耗存储资源和带宽，需要根据具体的迁移数据量，迁移数据能接受的耗时来定。评估方式为需要迁移的数据量/时间

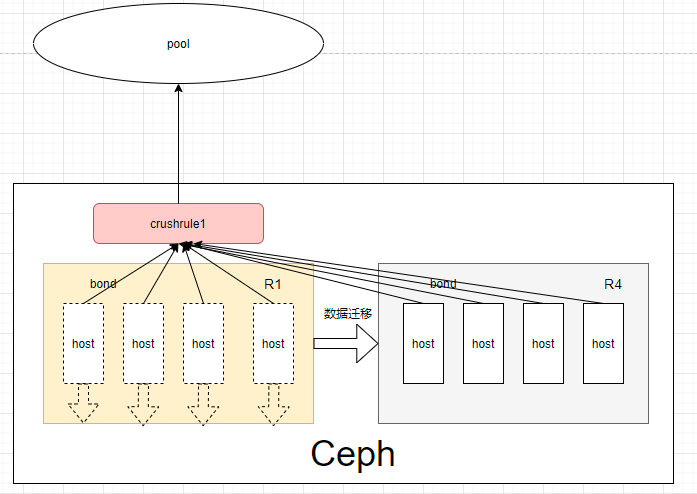
### 方案一

调整osd weight的方式迁移

1. 新增R4的服务器到Ceph集群
2. 调整R1的服务器的weight值为0，数据迁移完后，R1服务器下线

缺点：

1. pg二次计算，二次均衡。
2. 操作失败回滚会导致pg再次计算，再次数据均衡



### 方案二

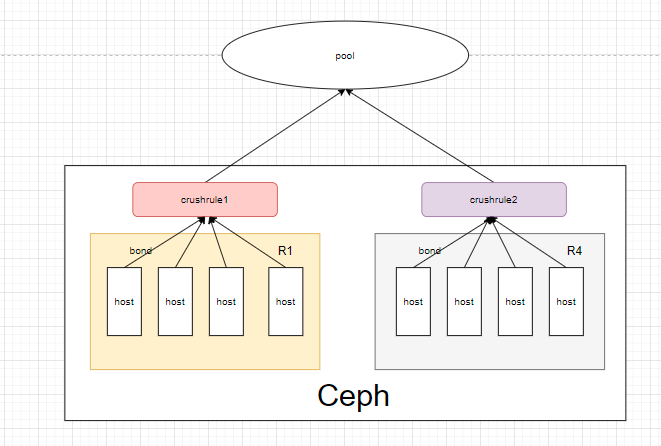
调整crushrule的方式

1. 新增R4的服务器到Ceph集群
2. R4的服务器新建一个crushrule
3. pool绑定到R4的crushrule，此时数据从R1迁出到R4，迁移完成后自动从R1删除数据

缺点：

操作失败回滚会导致pg再次计算，再次数据均衡

数据迁移过程中会锁住pg，对业务性能造成很大影响



### 方案三

使用crushrule+双倍副本的方式

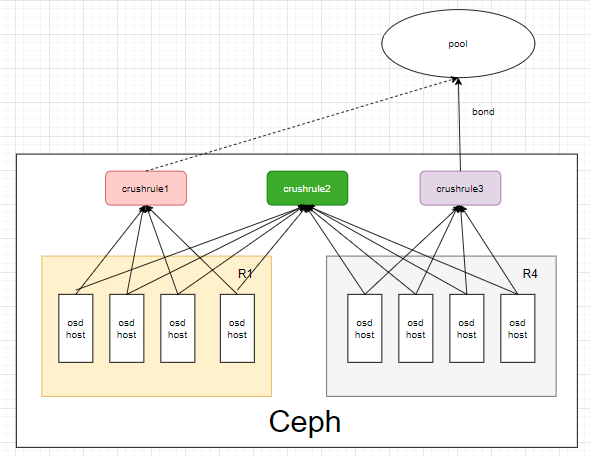
1. 新增R4的服务器到Ceph集群
2. R1的pool副本由2副本变成4副本 （此时会pg降级，不会造成数据均衡）
3. 新建crushrule2从R1和R4各选择2个副本，那crushrule2就是4副本。

(此时会从R1拷贝2个副本到R4，不是数据迁移，而是数据拷贝，R1中始终有2副本的数据)

1. 新建crushrule3在R4中选择2个副本
2. pool绑定crushrule3
3. R1的服务器下线

优点

1. 数据一致性复制即可，不需要二次数据均衡
2. 由于是数据复制而不是迁移，对集群性能影响较小
3. 每一步操作都可回滚，操作失败集群随时可以恢复到原来状态



### 迁移速度调整

逐步调整参数让osd的recover流量和业务的IO流量处于一个均衡状态

ceph tell osd.\* injectargs ‘–osd\_max\_backfills=3’

ceph tell osd.\* injectargs ‘–osd\_recovery\_max\_active=10’

ceph tell osd.\* injectargs ‘–osd\_recovery\_sleep=0.1’

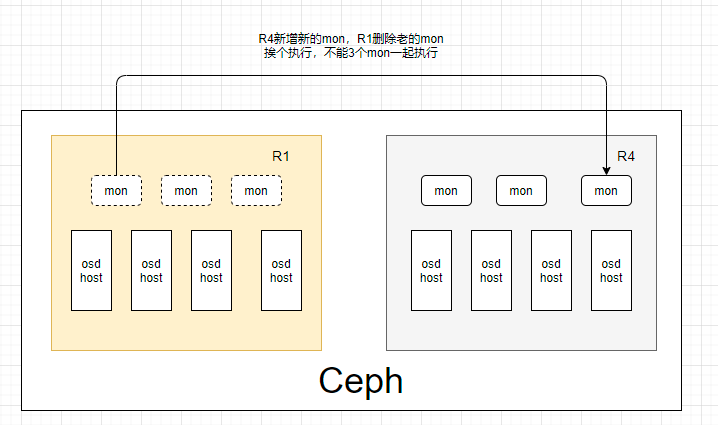
# monitor的迁移

monitor采用逐步迁移的方式

修改ceph.conf里的public\_network添加R4的网段地址

public\_network=192.168.0.1/24 ,192.168.0.2/24

R4新增一个monitor，R1删除一个monitor节点，直到R4有3个monitor节点，R1的monitor节点都删除了



### Cephfs的迁移

虽然客户端挂载了老的monitor地址，但是新增monitor后，新monitor的地址会以monmap的方式同步给cephfs客户端，所以客户端的链接依然有效不受影响

但是会影响新部署的pod，修改如下

1. 修改deployment/storageclass里的monitor ip地址
2. 如果是手动挂载的，mount命令需要修改成新的monitor ip地址

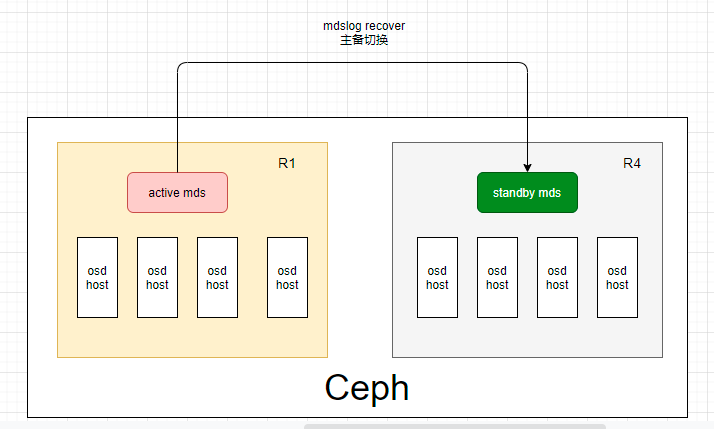
mount -t ceph 172.16.70.77:6789:/ /wyl -o name=admin,secret=AQBSdU5bT27AKxAAvKoWQdGpSuNRCHjC4B8DVA==

1. 如果是fstab挂载的，需要修改/etc/fstab文件里的ip地址为新的monitor ip地址

172.16.70.77:6789:/ /mnt/ceph ceph name=admin,secretfile=/etc/ceph/secret.key,noatime,\_netdev 0 2

mds的迁移

在R4新建一个standby mds，下线R4机房的active mds



### rbd的迁移

monitor新增后，会以monmap的方式同步给rbd客户端，所以rbd的客户端链接依然有效，业务不受影响

但是会影响新部署的pod，修改如下

修改deployment/storageclass里的monitor ip地址