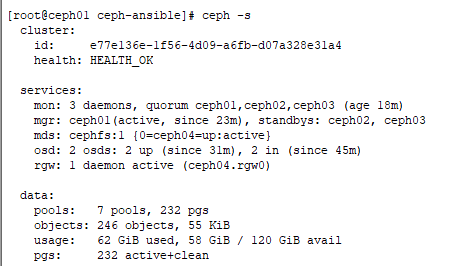
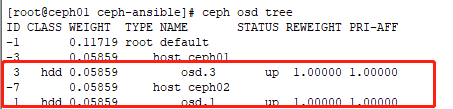
R1-R4数据迁移实验环境演练

### 准备R1的Ceph集群

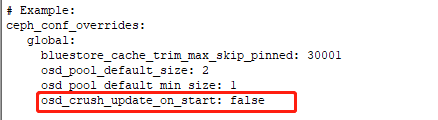
Ceph集群2副本





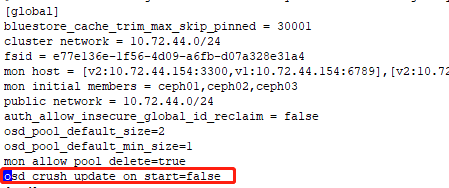
### 配置ceph-ansible的 group/all.yaml 文件

添加 osd\_crush\_update\_on\_start ：false



修改ceph节点的ceph.conf

添加 osd\_crush\_update\_on\_start= false

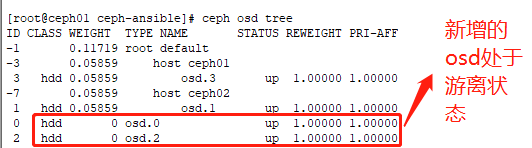


### 添加R4的服务器和硬盘到ceph集群

执行ceph-ansible

ansible-playbook infrastructure-playbooks/add-osd.yml

添加完后 osd显示游离状态，不属于root default这个device root



### 创建R4机房的设备树根

ceph osd crush add-bucket r4\_root root

ceph osd crush add-bucket ceph03 host

ceph osd crush add-bucket ceph04 host

ceph osd crush move ceph03 root=r4\_root

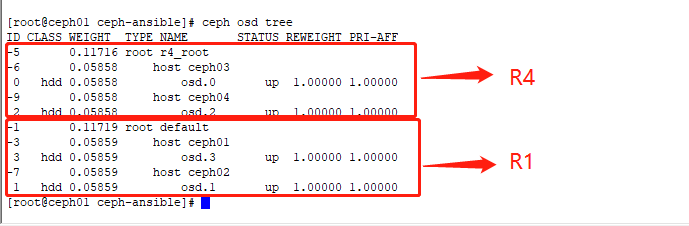
ceph osd crush move ceph04 root=r4\_root

ceph osd crush add osd.0 0.05859 host=ceph03

ceph osd crush add osd.2 0.05859 host=ceph04

0.05859 是osd weight 保持和r1的osd weight一致

创建完R4机房的设备树后如下



### 编辑cushmap

### 导出CRUSH MAP

ceph osd getcrushmap -o map

crushtool -d map -o map.txt

### 在map.txt 最后添加以下内容

vim map.txt

# rules

rule replicated\_ruleset {

id 0

type replicated

min\_size 1

max\_size 10

step take default

step chooseleaf firstn 0 type host

step emit

}

>>>>>>>>>>>>>> 添加开始 >>>>>>>>>>

rule replicated\_ruleset1 {

id 1

type replicated

min\_size 1

max\_size 10

step take default

step chooseleaf firstn 2 type host

step emit

}

rule replicated\_ruleset2 {

id 2

type replicated

min\_size 1

max\_size 10

step take default

step chooseleaf firstn 2 type host

step emit

step take r4\_root

step chooseleaf firstn 2 type host

step emit

}

rule replicated\_ruleset3 {

id 3

type replicated

min\_size 1

max\_size 10

step take r4\_root

step chooseleaf firstn 2 type host

step emit

}

<<<<<<<<<<<< 添加结束 <<<<<<<<<<<<

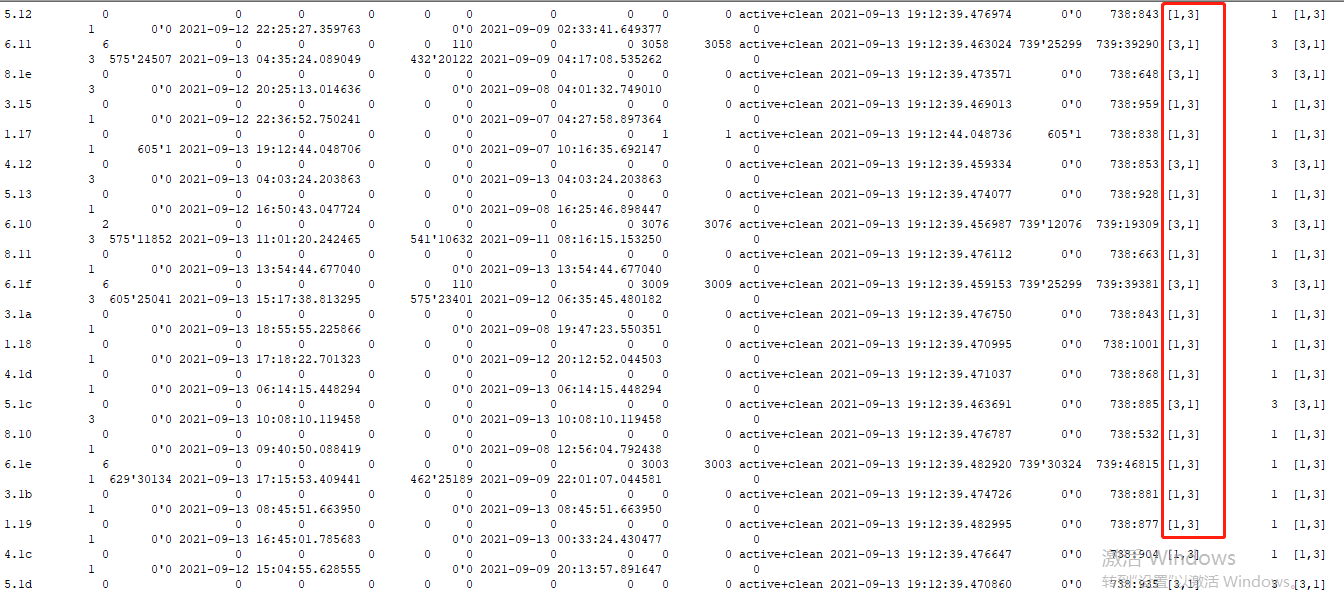
# end crush map

### 编译 CRUSH MAP，并注入到集群中

crushtool -c map.txt -o map.bin

ceph osd setcrushmap -i map.bin

此时数据没有任何迁移，数据都在R1。 R4的硬盘是空的，没有数据。

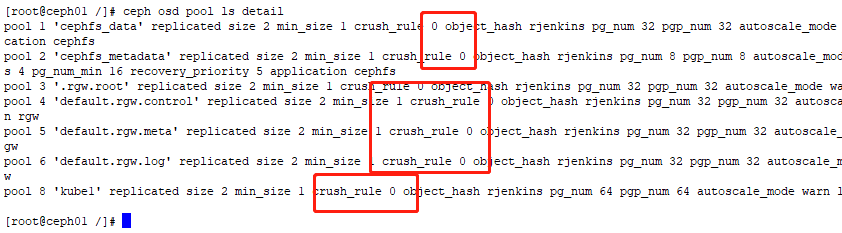


所有的数据都在osd.1和osd.3，也就是R1机房。

### 开始数据迁移

确认当前存储池的crushrule id=0

ceph osd pool ls detail



[root@ceph01 ~]# ceph osd pool set kube1 crush\_rule replicated\_ruleset1

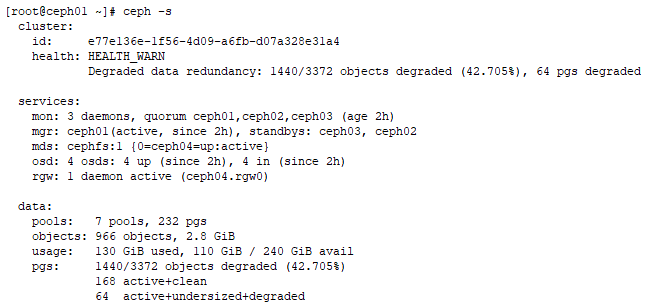
set pool 8 crush\_rule to replicated\_ruleset1

进行这个目的是一切后续的操作都基于ruleset1开始，如果迁移过程发生故障，可以回滚到ruleset0

将 kube1 池的 副本数设置为4， 副本设置成4后，因为原来R1机房只有2副本，所以副本是降级状态，变为 active+undersized+degraded。这个步骤为后续ceph集群在R4机房寻找另外2个副本做准备

[root@ceph cluster]# ceph osd pool set kube1 size 4

set pool 1 size to 4

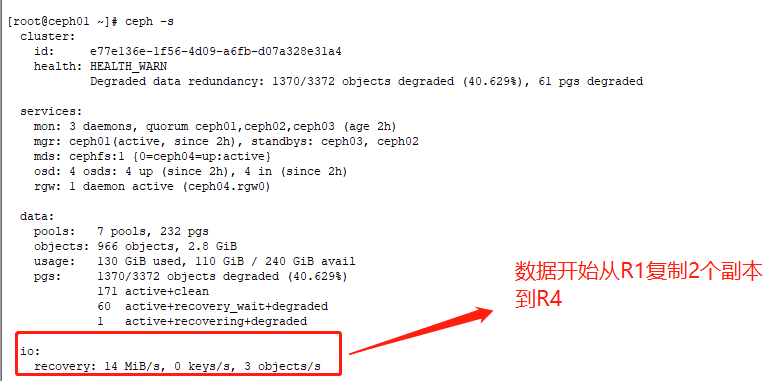


因为kube1这个池一共64个pg，所以有64个pg处于降级(degraded)状态

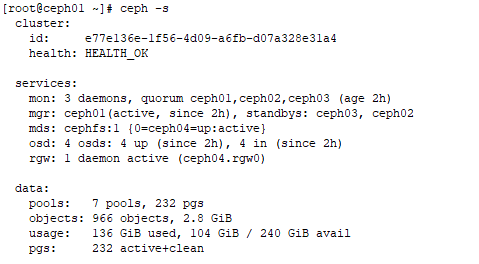
由于replicated\_ruleset2同时绑定了R1的设备树和R4的设备树，在R1中选择2个副本，在R4中选择2个副本，所以这步操作会导致数据复制，R1的2个副本，复制到R4的2个副本，使上一步kube1这个池的4副本达到正常状态

ceph osd pool set kube1 crush\_rule replicated\_ruleset2

set pool 8 crush\_rule to replicated\_ruleset2

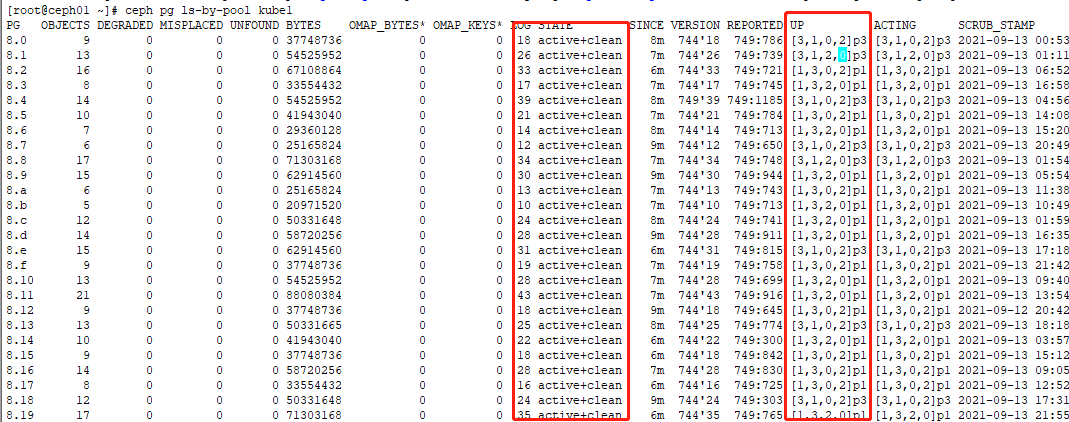


数据复制完后ceph集群恢复OK



并且kube1这个池的所有pg恢复健康状态。

我们可以看到kube1的所有pg都是4副本的，并且有2个副本是在osd.0和osd.2上，也即在R4机房里

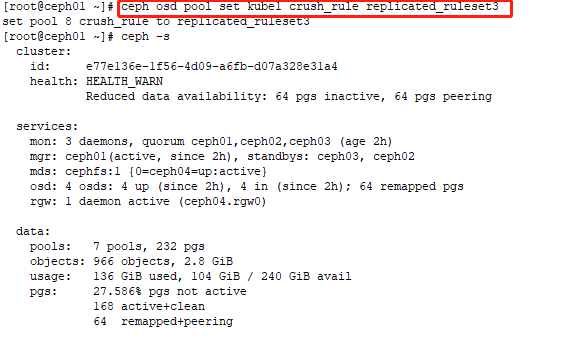


Crush rule 放弃R1上2个副本，选择R4上的2个副本

[root@ceph01 ~]# ceph osd pool set kube1 crush\_rule replicated\_ruleset3

set pool 8 crush\_rule to replicated\_ruleset3

此时副本又处于降级状态，因为kube1是4副本的，而replicated\_ruleset3只选择了R4里的2个副本。

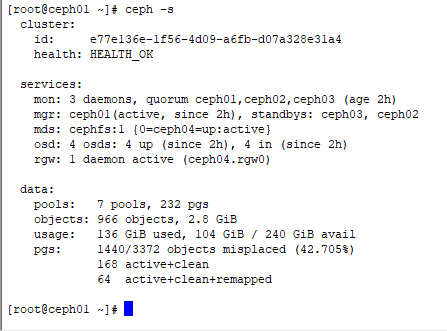


设置kube1为2副本，此时kube1抛弃了R1的2个副本，选择了R4的2个副本，R1的2个副本的数据被删除。

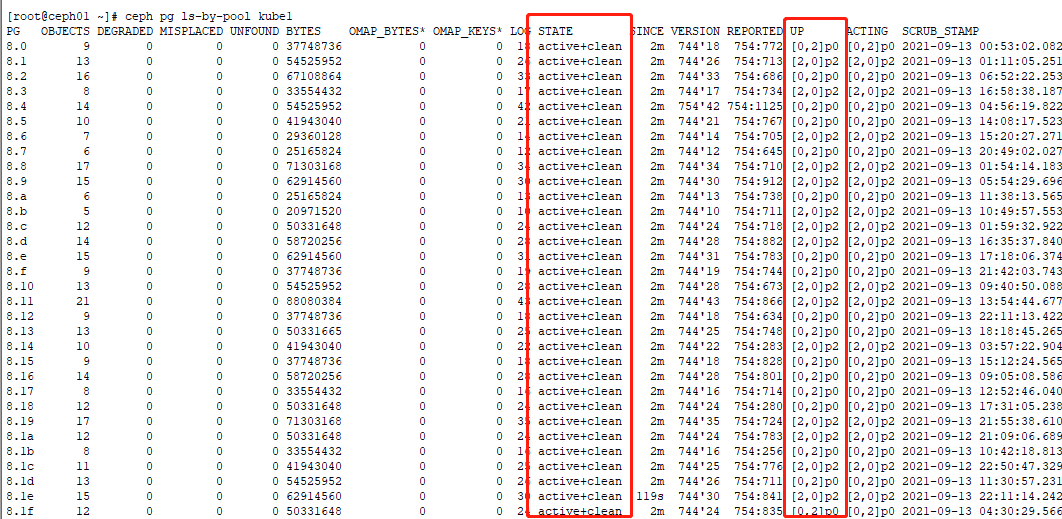
[root@ceph01 ~]# ceph osd pool set kube1 size 2

set pool 8 size to 2

ceph集群恢复正常



Kube1池变成2副本。pg数据全部转移到osd.0和osd.2。pg状态健康。



ceph集群删除R1的osd。下线R1的服务器。

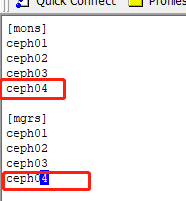
### monitor的迁移

实验monitor从R1迁移到R4

现在R4机房新增mon，再从R1机房删减mon。 挨个执行。

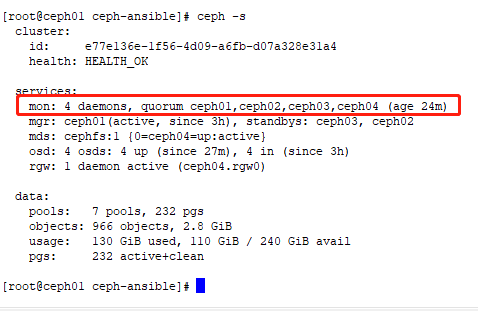
新增mon的步骤如下

在ceph-ansible的hosts文件里添加R4服务器作为mon



执行 ansible-playbook infrastructure-playbooks/add-mon.yml -i hosts

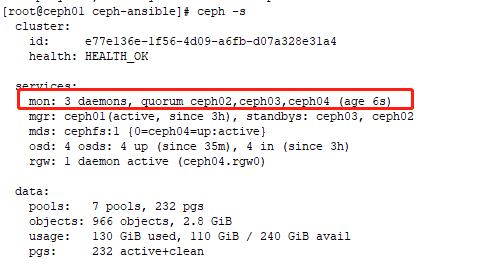
部署完后，集群有4个monitor



删除R1的monitor

ansible-playbook infrastructure-playbooks/shrink-mon.yml -e mon\_to\_kill=ceph01 -i hosts

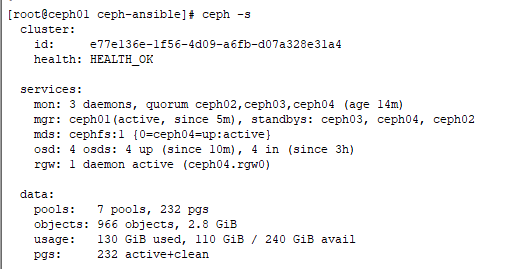
删除后mon为3个，R1机房中2个mon，R4机房中3个mon



### mgr 迁移

执行ansible-playbook site-container.yml -i hosts 向R4添加mgr，添加完后一共4个mgr。

R1机房有3个mgr，R4机房有1个mgr



删除R1机房的一个mgr，此时一共3个mgr，R1有2个mgr，R4有1个mgr

