实验环境手工配置crush map

参考文档

Copy of Ceph 集群Crush map初始化

操作过程

I. 修改当前crush map，为crush map添加新的type类型

> 注：该步骤是在对当前ceph集群中的crush map进行初始化操作之前进行的步骤；如果当前ceph集群的crush map已经初始化完成，需要对ceph进行扩容操作，则直接跳过该步骤！

1. 获取当前crush map的文本格式文件

```
ceph osd getcrushmap -o /tmp/old-map.bin        ## 获取crush map的二进制文件
crushtool -d map.bin -o /tmp/old-map.txt        ## 将crush map的二进制文件转换成文本文件
```

2. 手工编辑crush map的文本文件

```
vim /tmp/old-map.txt            ## 编辑文本文件，在文本文件中的type配置段中添加如下类型
>>
type 11 osd-domain
type 12 replica-domain
type 13 failure-domain
<<
```

3. 将crush map的文本文件导入并应用至当前的crush map当中

```
crushtool -c a.txt -o a.bin     ## 将文本格式的crush map文件保存为二进制格式
ceph osd setcrushmap -i a.bin         ## 将保存的二进制格式文件应用于当前的crush map中
```

II. 构建crush map的物理拓扑

1. 获取物理机架拓扑信息(由运维组提供)

```
rack-d01
10.0.101.68    ## 该信息表示IP地址为10.9.0.201的服务器位于名称为d07的物理机柜中
rack-d02
10.0.101.69
rack-d03
10.0.101.70
```

2. 检查当前集群中的osd是否运行正常

```
ceph osd dump | awk '{print $14,$15,$16,$17}' |grep " $NETWORK " | wc -l     ## 查看每个osd是否运行正常，
其中的$NETWORK表示osd监听socket的网络位，EX：10\.0\.101\.
```

3. 在crush map中对每个物理机架添加对应机架名称

应通过编辑crush map的文本文件来查看是否存在该机架名称

ceph osd crush add-bucket RACK rack　　## 在crush map中将物理机架添加至rack bucket，本例中三个RACK可设置为rack00，rack01，rack02

4. 通过命令行方式构建crush map中的物理拓扑

ceph osd crush move RACK root=default　　## 分别设置rack00~02三个机架的root为default

ceph osd crush add-bucket HOSTNAME host　　## 将指定主机的主机名称添加进host bucket中，
其中HOSTNAME是通过使用命令hostname查看得到的本机的主机名称

ceph osd crush move HOSTNAME rack= RACK　　## 将指定主机移动至指定机架，即在crush map中实现主机-->机架物理形式的映射

ceph osd cursh add OSDID　WEIGHT host= HOSTNAME　　## 将指定osd添加至指定主机，
即表示将每台osd主机中运行的多个osd实例进程添加至对应的osd主机中，其中OSDID是每个osd实例进程在osd集群中的id，EX：0,1,2等；
WEIGHT是每个osd进程实例在osd集群中的权重值，该权重W=该osd对应磁盘的容量M(GB)/1024；

此时就已经将crush map中的物理拓扑构建完成

III. 构建crush map的逻辑拓扑

1. 创建逻辑拓扑中的replica-domain(逻辑域)

replica domain可以实现replica domain中的多个osd domain之间进行数据复制

ssd类型磁盘的failure domain命名为failure-domain apple，sata类型磁盘的failure domain命名为failure-domain sata-xx

ceph osd crush add-bucket REPLICA-DOMAIN replica-domain　　## 在逻辑拓扑中添加replica domain bucket，
其中REPLICA-DOMAIN可以命名为replica-b-01

2. 创建osd-domain，每个replica-domain包含3个osd-domain

osd domain类似于物理拓扑中的主机，可以在每个osd domain中运行多个osd实例进程

ceph osd crush add-bucket OSD-DOMAIN osd-domain　　## 在逻辑拓扑中添加osd domain bucket，
其中OSD-DOMAIN可以命名为osd-group-b-01，osd-group-b-02，osd-group-b-03

3. 将osd进程实例添加至指定的osd-domain中

ceph osd crush add OSDID WEIGHT osd-domain= OSD-DOMAIN　　## 将指定的osd实例进程添加至指定的osd domain中，
其中OSDID和WEIGHT与物理拓扑中对应相同，本例中OSDID和OSD-DOMAIN的映射关系是osd.0~2从属于osd-group-b-01，
osd3~5从属于osd-group-b-02，osd6~8从属于osd-group-b-03

4. 将osd-domain放置到指定的replica-domain中

ceph osd crush move OSD-DOMAIN replica-domain= REPLICA-DOMAIN

5. 创建failure-domain

ceph osd crush add-bucket FAILURE-DOMAIN failure-domain

添加failure domain类型的bucket，ssd类型磁盘的的FAILURE-DOMAIN命名为failure-domain apple，
sata类型磁盘的的FAILURE-DOMAIN命名为failure-domain sata-XX

6. 将replica-domain放置到指定failure-domain

ceph osd crush move REPLICA-DOMAIN failure-domain=FAILURE-DOMAIN

7. 创建crush rule

```
## ssd类型磁盘的rule的名称设置为apple, ruleset设置为5
## sata类型磁盘的rule的名称设置为banana, ruleset设置为6
ceph osd getcrushmap -o old-crush-map.bin
crushtool -d old-crush-map.bin -o old-crush-map.txt
cp old-crush-map.txt new-crush-map.txt
vim new-crush-map.txt          ## 编辑crush map的文本格式文件, 添加对应的rule规则
>>
rule banana {      ## 配置rule名称为banana
   ruleset 6  ## 配置ruleset为6
   type replicated
   min_size 1
   max_size 10
   step take sata01  ## 指定rule规则: 指定rule应用的的failure domain
   step choose firstn 1 type replica-domain         ## 指定rule规则: 先选择replica-domain
   step chooseleaf firstn 0 type osd-domain         ## 指定rule规则: 再选择 osd-domain
   step emit
}
<<
crushtool -c new-crush-map.txt -o new-crush-map.bin
ceph osd setcrushmap -i new-crush-map.bin
```

此时就已经将crush map中的逻辑拓扑构建完成