## Redis持久化

**Redis配置**

使用redis前首先要弄明白是用来做缓存还是做存储系统，如果既做缓存有做存储，官方推荐使用2个Redis实例来完成

1.缓存（RDB）

把Redis当做一个缓存来用，所有的key都有过期时间，可以考虑 使用以下设置

maxmemory 2mb maxmemory-policy allkeys-lru

只要内存使用量到达2M，Redis就会使用类LRU算法自动删除某些key。

2.存储系统（推荐使用RDB和AOF两种持久化方式）

根据应用的持久化需求进行配置，并且只存储那些不需要被缓存的数据。

**Redis 持久化**

RDB和AOF两种方式，可以同时使用（先加载AOF文件），如果希望数据在服务器运行的时候存在,可以不使用任何持久化方式。

* RDB持久化方式能够在指定的时间间隔能对你的数据进行快照存储
* AOF持久化方式记录每次对服务器写的操作,当服务器重启的时候会重新执行这些命令来恢复原始的数据

RDB优点

1.快照（定时备份数据），使用单独子进程来进行持久化，主进程不会进行任何IO操作，保证了redis的高性能

2.单一文件存储，方便备份

3.恢复大的数据集的时，比AOF更快

RDB缺点

1.意外停止工作(比如停电)，会丢失快照间隔部分的数据

2.密集形成快照会影响redis的性能

配置见（SNAPSHOTTING）部分，此种方式是默认开启的，数据会持久化到dump.rdb文件

默认的快照

save 900 1

save 300 10

save 60 10000 ： 60 秒内有至少有 1000 个键被改动 快照一次

save "" 关闭 RDB 持久化

可以通过命令 save或 bgsave 立即完成数据的持久化

AOF优点

1.默认使用每秒文件同步策略，出现意外数据最多丢失1秒，更高数据的完整性

2.AOF文件比RDB文件大，且恢复速度慢

3.如果日志写入不完整支持redis-check-aof来进行日志修复

4.可以人工干预持久化的数据（打开aof文件，增删一些命令）

AOF缺点：

AOF文件比RDB文件大，且恢复速度慢。

参见 APPEND ONLY MODE 部分

AOF默认关闭，开启方法，appendonly 由 no 改为 yes

# appendfsync always 一直同步

appendfsync everysec 每秒同步

# appendfsync no 不同步

因为AOF是对aof文件追加命令，所以会越来越大，可以通过bgrewriteaof命令来完成aof文件的重写，通过配置文件可以配置aof文件达到多少M时重写， 重写后的新 AOF 文件包含了恢复当前数据集所需的最小命令集合

auto-aof-rewrite-percentage 100（当目前的AOF文件大小超过上一次重写文件大小的百分之几时进行重写，如果没有重启过，则以启动时的AOF文件大小为依据）

auto-aof-rewrite-min-size 64mb（允许重写的最小AOF文件大小）

关于redis数据的备份

* 创建一个定期任务（cron job）， 每小时将一个 RDB 文件备份到一个文件夹， 并且每天将一个 RDB 文件备份到另一个文件夹。
* 确保快照的备份都带有相应的日期和时间信息， 每次执行定期任务脚本时， 使用 find 命令来删除过期的快照： 比如说， 你可以保留最近 48 小时内的每小时快照， 还可以保留最近一两个月的每日快照。
* 至少每天一次， 将 RDB 备份到你的数据中心之外， 或者至少是备份到你运行 Redis 服务器的物理机器之外

## 基于两台服务器，3主3从，6个节点的redis集群

1.安装redis

<https://redis.io/download>

2.修改 redis.conf 配置

bind 192.168.136.130 #本机ip

port 7000 # 7000 7001 7002

daemonize yes

pidfile /var/run/redis\_7000.pid #7000 7001 7002

logfile "/usr/local/src/redis-cluster/7000/redis.log" #7000 7001 7002

dir /usr/local/src/redis-cluster/7000 #7000 7001 7002

appendonly yes

appendfilename "appendonly.aof"

appendfsync everysec

cluster-enabled yes

cluster-config-file nodes.conf

cluster-node-timeout 5000

3.

mkdir redis-cluster

cd redis-cluster

mkdir 7000 7001 7002

复制redis.conf 到7000 7001 7002 文件夹 并替换配置文件中所有和端口有关的项（如port 7000 # 7000 7001 7002 ）为对应的文件夹名

4.复制redis-server文件到 redis-cluster

5.另一台机器安装redis

6.重复2、3、4

7.安装ruby和rubygems（1.9后ruby标准库自带）

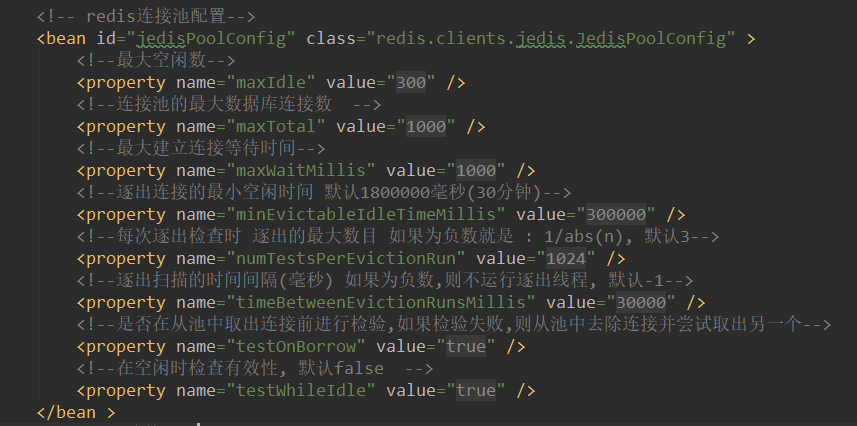
8.gem install redis

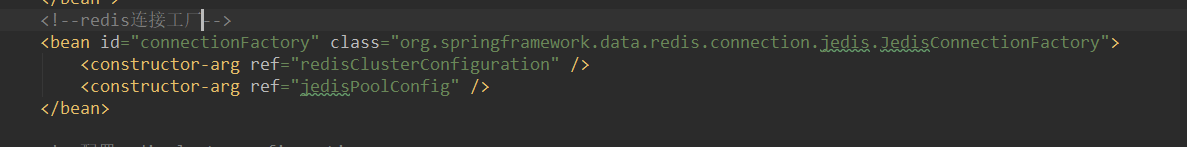
9.redis-trib.rb create --replicas 1 192.168.136.130:7000 192.168.136.130:7001 192.168.136.130:7002 192.168.136.128:7003 192.168.136.128:7004 192.168.136.128:7005

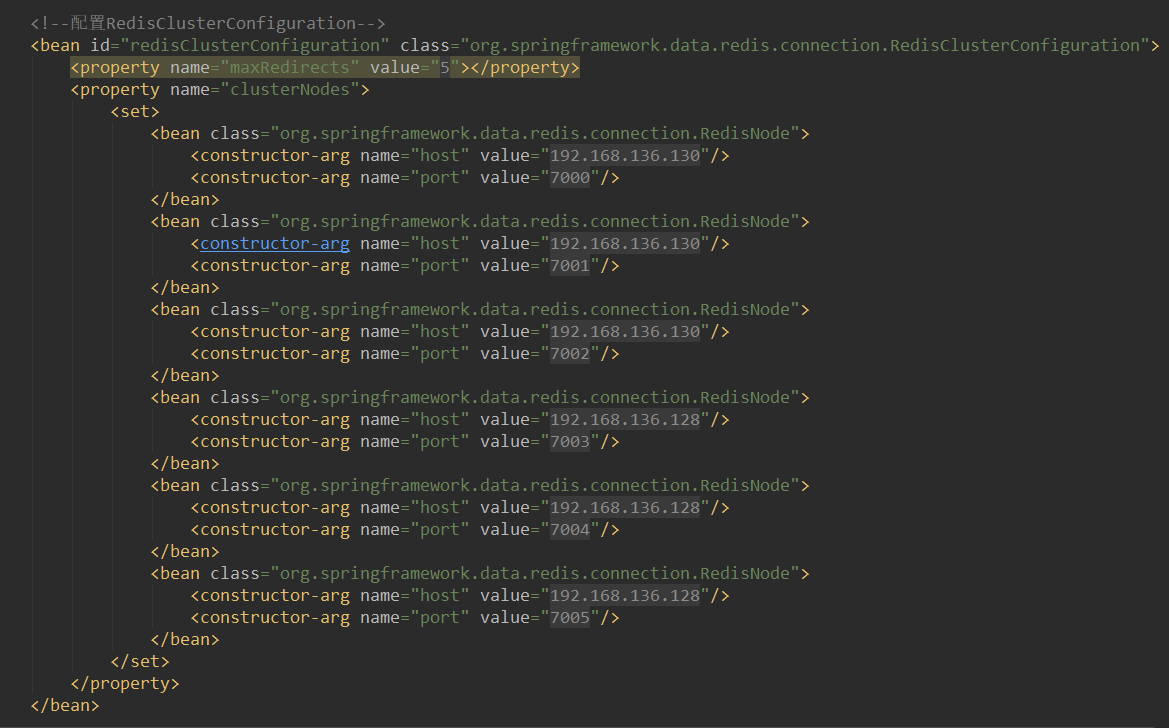
10.测试集群

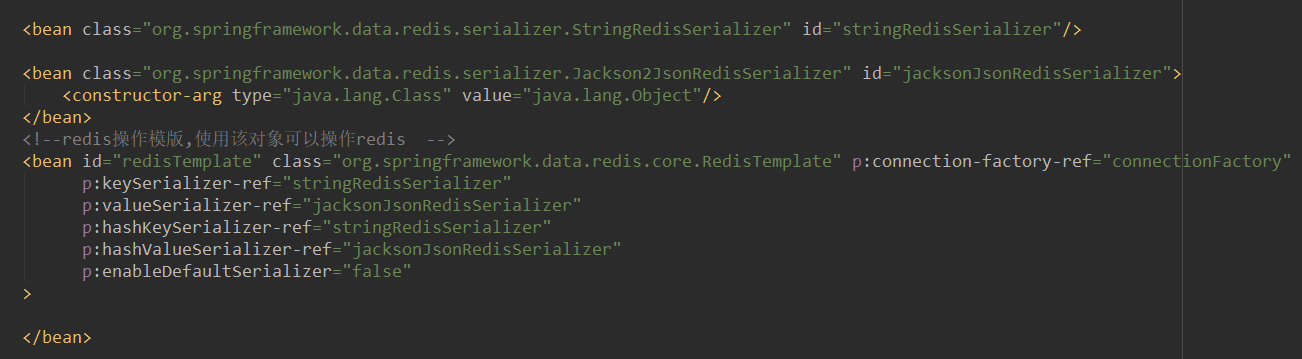
redis-cli -c -h 192.168.136.130 -p 7000

## Redis集群与Spring的整合











集群中查看结果

