Redis配置文件参数学习

进入 redis的安装目录，使用vim打开redis.conf vi redis.conf

#代表注释掉的内容，想让配置属性生效必须去掉前面的#然后重启redis

**后台运行**，yes是后台运行，no前台运行，将输出，输出到终端（默认）

**daemonize yes**

如果daemonize参数为yes的话就会产生pid文件，以下是pid文件的定义

pidfile /usr/local/redis-master/run/redis.pid

监听的端口

port 6379

绑定监听的IP地址

bind 127.0.0.1

如果在本地调用redis可以直接用sock文件

unixsocket /tmp/redis.sock //sock文件的位置

unixsocketperm 755 //sock文件的权限

如果一个链接在N秒内是空闲的，就将其关闭

timeout 0

如果对方down了或者中间网络断了发送ACK到客户端在指定的时间内没有收到对方的回应就断开TCP链接（时间单位秒记），此参数会受到内核参数的影响，推荐配置60。

tcp-keepalive 0

指定输出消息的级别

# debug (调试级别，详细信息，信息量大)

# verbose (详细信息，信息量较大)

# notice (通知，生产环境推荐)

# warning (错误信息警告信息)

loglevel notice

日志输出文件，默认在前端运行的时候此key的默认值是stdout输出到终端，如果用守护进程运行此key的stdout的时候将日志输入到/dev/null，如果想记录日志，就必须为其指定logfile位置

logfile /var/log/redis.log

将日志记录的哦syslog

syslog-enabled no

指定syslog的身份

syslog-ident redis

指定syslog的级别，必须是LOCAL0-LOCAL7之间

syslog-facility local0

设置数据库的数量。默认数据库DB 0，你可以选择一个不同的per-connection的使用SELECT<dbid>这儿的DBID是一个介于0和'databases'-1

databases 16

2.快照配置

将DB保存到磁盘的规则定义（快照）

格式：save <seconds> <changes>

save 900 1 //900秒内至少一个键值发生改变

save 300 10 //300秒内至少10个键值发声改变

save 60 10000 // 60秒内至少10000个键值发声改变、

如果启用如上的快照（RDB），在一个存盘点之后，可能磁盘会坏掉或者权限问题，redis将依然能正常工作

stop-writes-on-bgsave-error yes

是否将字符串用LZF压缩到.rdb 数据库中，如果想节省CPU资源可以将其设置成no，但是字符串存储在磁盘上占用空间会很大，默认是yes

rdbcompression yes

rdb文件的校验，如果校验将避免文件格式坏掉，如果不校验将在每次操作文件时要付出校验过程的资源新能，将此参数设置为no，将跳过校验

rdbchecksum yes

转储数据的文件名

dbfilename dump.rdb

redis的工作目录，它会将转储文件存储到这个目录下，并生成一个附加文件

dir /usr/local/redis-master/db

3.主从参数

如果本地是salve服务器那么配置该项

# slaveof <masterip> <masterport>

slaveof 127.0.0.1 65532

master的验证密码

masterauth <master-password>

当从主机脱离主的链接时，如果此值为yes当客户端查询从时，回响应客户端，如果是第一次同步回返回一个日期数据或这空值，如果设置为no，则返回“SYNC with master in progress”到INFO and SLAVEOF

slave-serve-stale-data yes

从服务器只读（默认）

slave-read-only yes

从发送ping到主的时间间隔(单位：秒)

repl-ping-slave-period 10

批量传输I/O超时和主数据或ping响应超时 默认60s 必须大于repl-ping-slave-period值

repl-timeout 60

此选项如果是“yes”那么Redis的使用数量较少的TCP数据包和更少的带宽将数据发送到，在从主机上延迟40毫秒（linux kernel中的40毫秒）出现。如果是no将在slave中减少延迟，但是流量使用回相对多一些，如果用多个从主机，此处建议设置成yes

repl-disable-tcp-nodelay no

从主机的优先级，如果当主主机挂了的时候，将从从主机中选取一个作为其他从机的主，首先优先级的数字最低的将成为主，0是一个特殊的级别，0将永远不会成为主。默认值是100.

slave-priority 100

redis达到maxmemory时，如何删除key

volatile-lru -> 用lru算法删除过期的键值

allkeys-lru -> 用lru算法删除所有键值

volatile-random -> 随机删除过期的键值

allkeys-random -> 随机删除任何键值

volatile-ttl -> 删除最近要到期的键值（监控TTL）

noeviction -> 不会写操作，返回一个错误

默认值例子如下：

maxmemory-policy volatile-lru

LRU的TTL算法是不准确，在几个中挑几个来检查近期用的最少的键值删除，为了节约内存可以设置小点。

maxmemory-samples 3