

## 一、redis的安装:

### 环境依赖:

```
1 一、调整文件句柄和用户进程数量:
2 1、vim /etc/sysctl.conf
3 vm.swappiness = 5 # 注释: 内存剩余 5% 的时候,
4 vm.dirty_ratio = 20 # 注释: 百分比值。当脏数据结
5 vm.dirty_background_ratio = 10 # 注释: 同上, 默认值为10。如果超过10%, 就会拍
6 net.ipv4.tcp_max_syn_backlog = 819200
7 net.core.netdev_max_backlog = 400000
8 net.core.somaxconn = 4096
9 net.ipv4.tcp_tw_reuse=1
10 net.ipv4.tcp_tw_recycle=0
11 vm.overcommit_memory = 1
12 [root@maxscale-test ~]# sysctl -p
13
14
15 二、vi /etc/security/limits.conf
16 # 文件产生时, 同样会生成文件句柄(文件存储单元与文件标识符的映射链), os会通过文件句
17 root soft nofile 65535
18 root hard nofile 65535
19 * soft nofile 65535
20 * hard nofile 65535
21
22
23 三、防火墙:
24 禁用selinux : /etc/sysconfig/selinux 更改SELINUX=disabled。
25 setenforce 0
26
27
28 iptables: 如果不使用可以关闭。如果使用则需要打开MySQL需要的端口号。
29 systemctl stop firewalld
30 systemctl disable firewalld
31
32
33
34 四、vim /etc/rc.local 在文件末尾添加如下指令:
35 if test -f /sys/kernel/mm/transparent_hugepage/defrag; then
36     echo never > /sys/kernel/mm/transparent_hugepage/defrag
37 fi
38 if test -f /sys/kernel/mm/transparent_hugepage/enabled; then
```

```
39 echo never > /sys/kernel/mm/transparent_hugepage/enabled
40 fi
```

## (一)、编译安装:

```
1 1、创建用户:
2 #groupadd -g 1010 redis
3 #useradd -u 1010 -g redis -s /sbin/nologin -d /home/redis redis
4
5 groupadd redis;useradd -g redis redis
6
7 2、创建环境目录:
8 mkdir -p /data/database/redis/archery_redis/{bin,conf,dump,logs,pid} ;cd /dat
9
10 3、解压目录:
11 cd /data/db_soft/;tar -zxvf redis-6.2.9.tar.gz
12
13 4、编译安装:
14 cd redis-6.2.9; make
15 make PREFIX=/usr/local/redis install
16 #指定目录make, 但是只有bin文件, config没有: make PREFIX=/usr/local/redis instal
17
18
19 5、拷贝文件bin文件和conf文件
20 cd /data/db_soft/redis-6.2.9;cp redis.conf sentinel.conf /data/database/redi
21 cd /data/db_soft/redis-6.2.9/src; cp mkreleasehdr.sh redis-trib.rb /usr/loc
22 # cd /data/db_soft/redis-6.2.9/src; cp mkreleasehdr.sh redis-benchmark redi
23
24
25
26 =====
27 # 推荐源码安装:
28 yum 安装:
29 1、使用 yum list 命令去看 yum 源中的 redis 安装包
30 [root@redis-m01 redis-6.2.9]# yum list|grep redis
31 Repodata is over 2 weeks old. Install yum-cron? Or run: yum makecache fast
32 collectd-redis.x86_64                    5.8.1-1.el7                                epel
```

```

33 collectd-write_redis.x86_64          5.8.1-1.el7          epel
34 hiredis.x86_64                       0.12.1-2.el7         epel
35 hiredis-devel.x86_64                 0.12.1-2.el7         epel
36 opensips-redis.x86_64                1.10.5-4.el7         epel
37 pcp-pmda-redis.x86_64                4.3.2-13.el7_9       update
38 php-nrk-Predis.noarch                1.0.4-1.el7          epel
39 php-pecl-redis.x86_64                2.2.8-1.el7          epel
40 php-phpiredis.x86_64                 1.0.0-2.el7          epel
41 python-trollius-redis.noarch          0.1.4-2.el7          epel
42 python2-django-redis.noarch           4.3.0-1.el7          epel
43 python2-redis.noarch                 2.10.6-2.el7         epel
44 python3-redis.noarch                 2.10.6-2.el7         epel
45 redis.x86_64                         3.2.12-2.el7         epel
46 redis-trib.noarch                    3.2.12-2.el7         epel
47 rubygem-redis.noarch                 3.2.1-2.el7          epel
48 rubygem-redis-doc.noarch              3.2.1-2.el7          epel
49 syslog-ng-redis.x86_64               3.5.6-3.el7          epel
50 uwsgi-logger-redis.x86_64            2.0.18-8.el7         epel
51 uwsgi-router-redis.x86_64            2.0.18-8.el7         epel
52
53
54 2、安装 yum install
55 yum install redis.x86_64
56
57 // 卸载:
58 yum remove redis
59
60 3、安装路径:
61 // 该路径是安装包里已经规定好的, 可以通过 find 命令查找。
62 find / -name redis
63 /etc/rc.d/init.d/redis
64 /etc/logrotate.d/redis
65 /var/lib/redis
66 /var/log/redis

```

### (三)、修改配置文件:

#### 1. 设置后台启动:

```
1 grep "daemonize" /data/database/redis/archery_redis/conf/redis.conf |grep -v
```

```

2 grep "pidfile" /data/database/redis/archery_redis/conf/redis.conf |grep -v "#"
3 grep "logfile" /data/database/redis/archery_redis/conf/redis.conf |grep -v "#"
4 grep "dir" /data/database/redis/archery_redis/conf/redis.conf |grep -v "#"
5 grep "port" /data/database/redis/archery_redis/conf/redis.conf |grep -v "#"
6
7 sed -i "s/daemonize no/daemonize yes/g" /data/database/redis/archery_redis/conf/redis.conf
8 sed -i "s/pidfile .*/pidfile \data\database\redis\archery_redis\pid\redis_6397.pid/g" /data/database/redis/archery_redis/conf/redis.conf
9 sed -i "s/logfile .*/logfile \data\database\redis\archery_redis\logs\redis_6397.log/g" /data/database/redis/archery_redis/conf/redis.conf
10 sed -i "s/dir .*/dir \data\database\redis\archery_redis\dump/g" /data/database/redis/archery_redis/conf/redis.conf
11

```

## 2、redis的最小配置：

```

1 // 最小配置：
2 port 6379
3 daemonize yes
4 bind 10.10.30.197
5 protected-mode yes
6 pidfile /data/database/redis/archery_redis/pid/redis_6397.pid
7 logfile /data/database/redis/archery_redis/logs/redis_6397.log
8 dir /data/database/redis/archery_redis/dump
9 dbfilename dump_6397.rdb
10 appendfilename "6397.aof"
11 requirepass 123456
12
13
14
15
16
17 // 网易线上的配置
18 daemonize yes
19 logfile redis.log
20 loglevel notice
21 pidfile redis.pid
22 port 6407
23 dir /redis/redis6407
24 dbfilename dump6407.rdb
25 maxmemory 10000mb
26 maxmemory-policy volatile-lru
27 client-output-buffer-limit slave 0 0 0
28 tcp-backlog 511
29 slave-priority 100

```

```
30 rename-command flushdb 163flushdb
31 rename-command flushall 163flushall
32 rename-command shutdown 163shutdown
33 rename-command KEYS 163keys
34 rename-command MONITOR 163monitor
35 rename-command SAVE 163save
36 bind 10.10.30.197
37 cluster-enabled yes
38 cluster-enabled yes
39 cluster-config-file "nodes6407.conf"
40 cluster-migration-barrier 100
41 cluster-node-timeout 15000
```

### 3、授权目录权限：

```
1 chown redis.redis /data/database/redis -R
```

#此处可以把redis放在固定的应用程序的目录里，然后在做软连接。

```
1 # 结合自己的环境选择执行：
2 ln -s /data/database/redis/archery_redis/bin /usr/local/redis
```

### (四)、su到redis目录：

```
1 su - redis
2 vim ~/.bash_profile
3 export PATH=/usr/local/redis/bin:$HOME/bin:$PATH
4 source ~/.bash_profile
```

## (五)、指定配置文件，启动redis服务：

```
1 /usr/local/redis/bin/redis-server /data/database/redis/archery_redis/conf/re
```

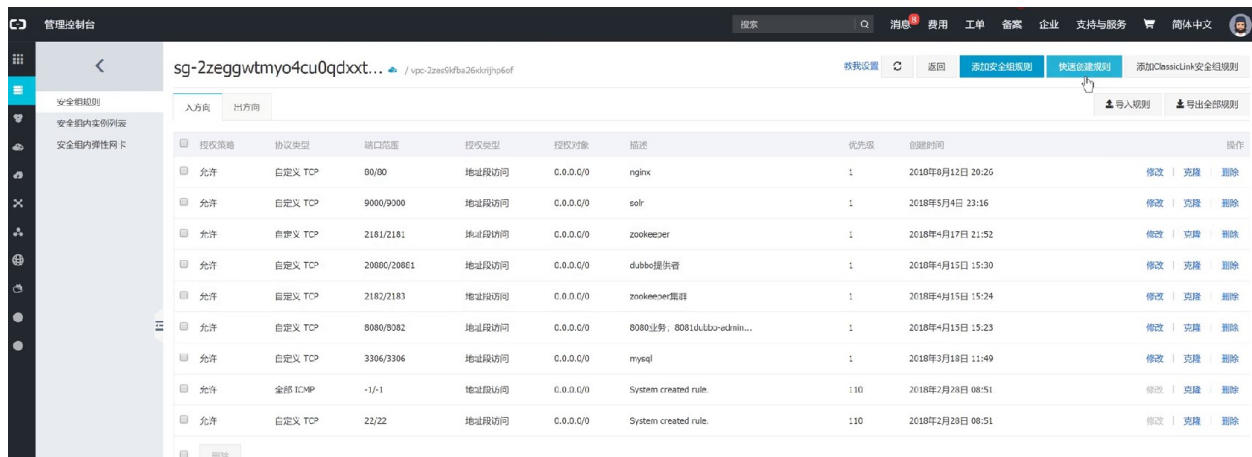
## (六)、查看redis的端口

```
1 ps -ef |grep redis
2
3 netstat -nalp|grep 6379
```

## (七)、客户端连接：

```
1 /usr/local/redis/bin/redis-cli
2
3 // 相关参数：
4 1、-c： 是以集群方式登录。
5 2、-h： 要登录主机的IP地址。 --> 不输入就是 127.0.0.1
6 3、-p： 要登录主机的端口号。 --> 默认 6379
7 4、-a： 要登录主机的 redis-server 的密码
8 5、--raw： 支持中文处理
9
10 // 线上登录方式：
11 redis-cli -h 127.0.0.1 -p 6379 -a 123456 --raw
12
13
14 =====
15 使用第三方工具登录 redis-server：
16 // 以下 3 个缺一不可：
17 1、开端口
18 2、修改 bind 属性
19 3、修改 protected-mode
20
21 如下图：
```

1、登录 阿里云的管理控制台 --> 安全组规则 --> 快速创建规则



### 快速创建规则

网卡类型: 内网

规则方向: 入方向

授权策略: 允许

常用端口(TCP): ☐ SSH (22) ☐ telnet (23) ☐ HTTP (80) ☐ HTTPS (443) ☐ MS SQL (1433) ☐ Oracle (1521) ☐ MySQL (3306) ☐ RDF (3389) ☐ PostgreSQL (5432) ☐ Redis (6379)

自定义端口: ☒ TCP ☐ UDP  ?

授权类型: 地址段访问

\* 授权对象:  ? 教我设置

优先级:  ?

描述:    
 长度为2-256个字符, 不能以http://或https://开头。

- 1、如果使用的是 ECS 则需要再 "安全规则" 中设置。
- 2、如果是用的 cenos 6.xz 系列, 需要开启 IPtables 防火墙中的 redis-server 端口。
- 3、如果是 cenos7.x 系列, 需要开启 firewall 防火罩中的 redis-server 的端口。

#### (八)、停止redis示例:

```
1 1、使用是shutdown 命令:
2 /usr/local/redis/bin/redis-cli shutdown
3
4 2、使用 pkill 命令: (不要 -9)
5 pkill redis-server
6
7 3、使用 kill
8 kill redis进程号
9
10 4、使用 kill -9 会丢失数据, 不会持久化。
11 kill -9 redis进程号
```

#### 警告:

vim sysctl.conf

vm.overcommit\_memory = 1

net.core.somaxconn= 1024

sysctl -p



=====

===

## 二、bin 目录文件:

```
1 // 命令文件:
2 1、redis-benchmark: 性能测试工具, 模拟 N 个客户端发出的 M 个请求。
3 2、redis-check-aof: AOF 持久化文件的修复工具。
4 3、redis-check-rdb: RDB 持久化文件修复工具。
5 4、redis-cli: 客户端程序。
6 5、redis-sentinel: 哨兵机制, 主要作用: 监听、选主、通知。
7 6、redis-server: 服务端程序
8 7、redis-trib.rb: 是 rubu 写的一个集群工具
9
10 // 配置文件:
11 1、redis.conf: 主配置文件, 出厂设置。
12 2、sentinel.conf: 哨兵配置文件。在执行 redis-sentinel 的时候应用该配置文件。
```

=====

===

## 三、redis的配置

vim redis.conf

```
1 ### 基础环境相关的配置:
2 daemonize yes
3 port 6380
4 masterauth "1hbl3qt!"
5 requirepass "1hbl3qt!"
6 pidfile "/app/redis_cluster/redis_6379/redis.pid"
7 logfile "/app/redis_cluster/redis_6379/redis.log"
8 dir "/app/redis_cluster/redis_6379/dump"
9 # 日志的级别, notice表示记录除了 debug 外的大部分信息
10 loglevel notice
11 # 使用最大的数据库数量。
12 databases 16
13 # 启动是否显示 redis 标志和版本信息。
14 always-show-logo yes
15 # 只接受来自 localhost 的连接, 默认为 yes。
16 protected-mode yes
17 bind 172.17.102.66 127.0.0.1
```

```
18
19
20
21
22 ### 内存相关的配置：（配置有问题）
23 # redis 的内存限制，默认为 0 不限制。
24 maxmemory 0
25
26 # 内存的 8 中淘汰策略：（参数有问题）
27 # 1、noeviction：不会被淘汰数据，一旦内存写满，则直接返回错误，并终止读写服务。
28 # 2、volatile-random：从已设置过期时间的数据集中随机淘汰数据。
29 # 3、volatile-ttl：从已设置过期时间的数据集中淘汰剩余生存时间（TTL）最短的数据。
30 # 4、volatile-lru：从已设置过期时间的数据集中淘汰最近、最少使用的数据。
31 # 5、volatile-lfu（Redis 4.0 后新增）：按照最近、最不常使用算法，清理设置了过期时
32 # 6、allkeys-lru：从所有数据中淘汰最近最少使用的数据。
33 # 7、allkeys-random：从所有数据中随机淘汰数据。
34 # 8、allkeys-lfu（Redis 4.0 后新增）：最近最不常使用的算法，清除所有 key 中最近最
35 maxmemory-policy noeviction
36
37
38
39
40 ### TCP 连接与进程相关的配置：
41 # tcp 的连接数，即redis 服务器接受新的连接请求之前，操作系统可以挂起的最大连接数。
42 tcp-backlog 511
43 # 每个 300 秒检查连接是否有效。有助于发现死链接并释放资源。
44 tcp-keepalive 300
45 # 客户端连接的超时时间，如果设置为 0，表示不超时。
46 timeout 0
47 # redis 是否使用外部守护进程（如 system upstart）来管理其进程。如果设置为 no 表示
48 supervised no
49 # 限制同时连接到 redis 服务器的客户端数量。
50 maxclients 50000
51
52
53
54
55 ### rename command 危险命令别名：
56 rename-command flushdb 163flushdb
57 rename-command flushall 163flushall
58 rename-command shutdown 163shutdown
59 rename-command KEYS 163keys
60 rename-command MONITOR 163monitor
```

```
61 rename-command SAVE 163save
62 rename-command EVAL 163eval
63
64
65
66 # RDB 相关的参数配置
67 # 当后台保存（bgsave）出错时，是否停止写入命令。yes：拒绝写入，no：继续写入。
68 stop-writes-on-bgsave-error no
69 # 是否启动内存碎片整理，可以提高内存使用效率。
70 activedefrag yes
71 # 是否在 RDB 创建时启用压缩，yes 表示启动，这可能会增加 cpu 的负载。
72 rdbcompression yes
73 # 是否在 RDB 文件中包含数据校验和，确保 RDB 文件在传输或存储过程中未被破坏
74 rdbchecksum yes
75 dbfilename "dump.rdb"
76 # RDB 触发条件：
77 # 900 秒内有 1 个 key 被更改，则触发 RDB 快照。
78 save 900 1
79 # 300 秒内有 10 个 key 被更改，则触发 RDB 快照。
80 save 300 10
81 # 60 秒内有 10000 个 key 被更改，则触发 RDB 快照。
82 save 60 10000
83
84
85
86 # 主从复制相关参数配置：
87 # 设置为 yes 时，表示主从断开后，从库仍然能继续提供数据服务，即使数据不是最新状态。
88 replica-serve-stale-data yes
89 # 设置为 yes 表示 从节点为只读，方式从节点做写入操作。
90 replica-read-only yes
91 # 是否允许无磁盘同步，如果设置为yes，表示从库可以在不执行完整的 RDB 快照同步的情况下
92 repl-diskless-sync no
93 # 当使用无磁盘同步时，这个参数指定了从库在开始同步之前等待的时间（以秒为单位）
94 repl-diskless-sync-delay 5
95 # 是否禁用 TCP_NODELAY 套接字选项，禁用此选项可以减少网络延迟，但是会增加每次传输的
96 repl-disable-tcp-nodelay no
97 # 设置主从故障切换时，升级主节点的优先级。
98 replica-priority 100
99 # 设置从节点的优先级。
100 slave-priority 100
101 # 设置复制操作的超时时间（以秒为单位）。在这个时间内如果没有收到主节点的数据，从节点
102 repl-timeout 60
103 # 设置至少有多少从库的连接处于良好状态时，主节点才会执行写操作。这有助于确保主从数据
```

```
104 min-slaves-to-write 1
105 # 设置主从复制延迟的最大时间（以秒为单位），如果从节点超过这个值，主节点将停止接收写
106 min-slaves-max-lag 10
107 # 设置从节点客户端输出缓冲区的限制，
108 # config set client-output-buffer-limit slave 512mb 128mb 60
109 # 第一个 slave 参数表明该配置项是针对复制缓冲区的（注意：高版本 slave 变成 replic
110 # 512mb 代表将缓冲区大小的上限设置为 512MB
111 # 128mb 和 60 代表的设置是，如果连续 60 秒内的写入量超过 128MB 的话，也会触发缓冲
112 client-output-buffer-limit slave 0 0 0
113
114
115
116 # 惰性异步删除相关的参数：
117 # 内存达到maxmemory并设置了淘汰策略时尝试异步释放内存，yes：表示后台线程中异步地释
118 lazyfree-lazy-eviction no
119 # key在过期删除时尝试异步释放内存，yes：表示会在后台线程中异步地处理过期键的内存释
120 lazyfree-lazy-expire no
121 # 执行RENAME/MOVE等命令或需要覆盖一个key时，删除旧key尝试异步释放内存，yes：则删除
122 lazyfree-lazy-server-del no
123 # 主从全量同步，从库清空数据库时异步释放内存。yes：表示从节点执行全量同步时，会异步
124 replica-lazy-flush no
125
126
127
128 # AOF 重写配置相关参数：
129 # 是否开启 AOF 持久化
130 appendonly no
131 appendfilename "appendonly.aof"
132 # AOF 的同步频率：
133 appendfsync everysec
134 # AOF 重写时是否禁用 Appendfsync。yes：重写期间不同步AOF，提高重写性能，但会丢失数
135 no-appendfsync-on-rewrite no
136 # 触发自动AOF 重写的条件，这里是当 AOF 文件大小比上次重写后的大小增加 100% 时触发重
137 auto-aof-rewrite-percentage 100
138 # 设置 AOF 重写的文件大小限制，当达到或者超过 64MB 时，才会根据 auto-aof-rewrite-
139 auto-aof-rewrite-min-size 64mb
140 # redis 重启时，AOF 文件被截断，是否允许redis 加载并恢复数据，设置为 yes 可以提高
141 aof-load-truncated yes
142 # 在 AOF 重写是，是否使用RDB 格式的前缀。如果设置为 yes，AOF 文件将首先包含一个 R
143 aof-use-rdb-preamble no
144 # 在 AOF 重写 过程中，是否使用增量式文件同步。如果设置为 yes，可以减少磁盘I/O操作，
145 aof-rewrite-incremental-fsync yes
146
```

```
147
148
149 # lua 脚本配置相关参数:
150 # 指定 lua 脚本执行的最大时间(以毫秒为单位), 如果脚本运行时间超过这个限制, redis 将
151 lua-time-limit 5000
152
153
154
155
156 # 慢日志相关配置参数:
157 # 设置慢日志记录的阈值。只有当查询或命令的执行时间超过这个阈值(以微妙为单位)时, 才
158 slowlog-log-slower-than 10000
159 # 设置慢查询日志的最大长度。这是慢查询日志可以保存的条目数量。当达到这个长度时, 最早
160 slowlog-max-len 128
161 # latency monitor,default off
162 # 设置延迟监控的阈值, 如果设置为非零值, redis 将监控并记录所有超过这个阈值(以微妙)
163 latency-monitor-threshold 0
164
165
166 # notify,default off
167 # 设置发布订阅机制需要被通知的事件类型。空字符串 "" 表示不监听任何事件。这个参数可
168 notify-keyspace-events ""
169
170
171
172 #### 用于控制 redis 中不同数据结构的内存使用和存储效率:
173 # 设置 hash 表中使用 ziplist(压缩列表)格式存储的最大条目数, 当哈希表的字段超过这
174 hash-max-ziplist-entries 512
175 # 设置 ziplist(压缩列表)格式的每个字段值的最大长度(以字节为单位)。当字段值超过
176 hash-max-ziplist-value 64
177 # 设置列表(list)使用 ziplist 格式的最大内存大小。-2: 表示使用默认大小; -1: (使用
178 list-max-ziplist-size -2
179 # 设置列表压缩的深度, 列表压缩尝试减少内存使用, 但会增加 cpu 开销。这里设置为 0 表
180 list-compress-depth 0
181 # 设置集合(Set) 中使用整数集合(intset)格式存储最大条目数。当集合中的元素超过这
182 set-max-intset-entries 512
183 # 设置有序集合(sorted set) 中使用 ziplist 格式存储的最大条目数, 当有序集合的元素
184 zset-max-ziplist-entries 128
185 # 设置有序集合中 ziplist 格式的每个成员的最大长度(以字节为单位)。当成员长度超过这
186 zset-max-ziplist-value 64
187 # 设置 HyperLogLog 稀疏表示形式的最大字节数。当 HyperLogLog 对象的大小超过这个值时
188 hll-sparse-max-bytes 3000
189 # 设置 是否启用动态哈希。yes: redis 会在运行时重新哈希哈希表(rehash 操作), 以减
```

```

190 activerehashing yes
191
192
193 ### 用于限制客户端输出缓冲区的大小，避免客户端因为无法发送数据而导致内存消耗过多。
194 # 针对普通客户端（normal）
195 # 第一个 0 表示硬限制，即输出缓冲区最大允许大小。
196 # 第二个和第三个 0 表示软性限制（soft limit）和超时时间。设置为 0 表示不设置软性限
197 client-output-buffer-limit normal 0 0 0
198
199 # 针对从节点（副本）客户端的设置：
200 # 256mb：是硬性限制，表示从节点输出缓冲区最大不能超过256兆字节。
201 # 64mb：是软性限制，表示当缓冲区达到 64 兆字节时，Redis 会采用一些措施，如暂停读取
202 # 60：表示如果超过软性限制的大小，达到 60 秒时，redis 将关闭连接。
203 client-output-buffer-limit replica 256mb 64mb 60
204
205 # 针对发布/订阅（pub/sub）客户端的设置。
206 # 32mb：是硬性限制，表示 pub/sub 客户端的输出缓冲区最大不能超过 32 兆字节。
207 # 8mb：是软性限制，表示当缓冲区达到 8 兆字节时，redis 会采用一些措施。
208 # 60：表示如果超过软性限制达到 60 秒时，redis 将关闭连接。
209 client-output-buffer-limit pubsub 32mb 8mb 60
210
211
212 ### 设置 redis 用来控制多种操作的时间轮询频率（单位是每秒多少次），例如：数据库的清
213 ### 注意：hz 的值越高，这些操作被执行的频率就越高，但同时也会相应增加 CPU 的使用率
214 ### 值 10 表示每秒钟进行 10 次轮询，这是一个相对较低的频率，适用于低负载环境。
215 hz 10
216
217
218 ### 设置当前 redis 实例做为从节点（副本），并制定需要同步的主节点的 IP + 端口。
219 # 172.17.102.67：是主节点的 IP 地址， 6379：是服务器默认端口号
220 # （配置这个参数后，当前 redis 实例将尝试连接到指定的主节点，并开始数据复制过程）
221 # replicaof 172.17.102.67 6379
222

```

=====

## redis-cli命令详解：

### 1、-x

从标准输入读取一个参数

例如：