

## 二、Redis多实例部署

单实例单线程的redis进程不足以高效率使用cpu和内存资源，所以一般来讲redis在同一台机器上要启动多个进程完成多实例部署；默认占用6379的情况下无法直接的3个实例启动，这里我们需要了解如何通过制定配置文件，将多实例部署在linux上

启动redis服务的命令redis-server，没有加载任何配置文件指定各种各样的配置信息（端口指定，ip绑定，后台运行）

例如在根目录存在一个配置文件的模板（大部分与默认启动的配置相同），即redis.conf  
配置文件（在启动时指定的配置文件，核心的配置文件）

/redis根目录/redis.conf

```
1 root root 92766 Sep 21 22:20 00-RELEASENOTES
1 root root 53 Sep 21 22:20 BUGS
1 root root 1805 Sep 21 22:20 CONTRIBUTING
1 root root 1487 Sep 21 22:20 COPYING
x 7 root root 4096 Jan 16 16:15 deps
1 root root 273 Jan 17 10:33 dump.rdb
1 root root 11 Sep 21 22:20 INSTALL
1 root root 151 Sep 21 22:20 Makefile
1 root root 4223 Sep 21 22:20 MANIFESTO
1 root root 6834 Sep 21 22:20 README.md
1 root root 46695 Sep 21 22:20 redis.conf
x 1 root root 271 Sep 21 22:20 runtest
x 1 root root 280 Sep 21 22:20 runtest-cluster
x 1 root root 281 Sep 21 22:20 runtest-sentinel
1 root root 7606 Sep 21 22:20 sentinel.conf
x 2 root root 4096 Jan 16 16:16 src
x 10 root root 4096 Sep 21 22:20 tests
x 7 root root 4096 Sep 21 22:20 utils
9-62-65 redis-3.2.11]#
```

```
# Note on units: when memory size is needed, it is possible to :
# it in the usual form of 1k 5GB 4M and so forth:
#
# 1k => 1000 bytes
# 1kb => 1024 bytes
# 1m => 1000000 bytes
# 1mb => 1024*1024 bytes
# 1g => 1000000000 bytes
# 1gb => 1024*1024*1024 bytes
#
# units are case insensitive so 1GB 1Gb 1Gb are all the same.
```

一个redis实例默认占用所有物理内存（上限是物理内存大小），在实际使用中需要限制大小  
配置文件的修改内容

p61 bind 用#注释掉

```
#
# IF YOU ARE SURE YOU WANT YOUR INSTANCE
# JUST COMMENT THE FOLLOWING LINE.
# ~~~~~
#bind 127.0.0.1
```

如果需要绑定监听的ip（客户端只有通过被绑定的ip才可以利用redis-cli -h ip地址链接服务器）

bind 127.0.0.1 106.75.101.219（外网可访问当前服务器的ip）

bind一旦使用#注释掉，没有任何的限制，都可以链接服务器（初学者建议注释掉）

p80 保护模式不启动

```
75 #
76 # By default protected mode is enabled. You should d
77 # you are sure you want clients from other hosts to
78 # even if no authentication is configured, nor a spe
    aces
79 # are explicitly listed using the "bind" directive.
80 protected-mode no
```

保护模式开启，需要登录密码，教学环境改成no

p84 6379是默认端口（要启动其他的redis实例需要修改端口）

```
44).
83 # If port 0 is specified Redis will not listen on a TCP socket.
84 port 6379
85
86 # TCP listen() backlog
```

p105 当客户端空闲时间达到一小时，就会自动断开连接，0秒表示不启用超时配置

```
103
104 # Close the connection after a client is idle for N seconds (0 to disab
    le)
105 timeout 3600
106
```

p128 daemonize 设置成yes让redis服务器启动有守护进程管理  
（后台执行）

```
125
126 # By default Redis does not run as a daemon. Use 'yes' if you
127 # Note that Redis will write a pid file in /var/run/redis.pid
    onized.
128 daemonize yes
129
```

p150 对应不同的redis实例，pid的文件名称需要和端口同名的每个进程在linux或者其他操作系统中都会占用pid号，当系统中的进程较多时，需要查找redis进程号可能叫麻烦，直接打开pid文件查看

```
147 #
148 # Creating a pid file is best effort: if Redis is not able to cr
149 # nothing bad happens, the server will start and run normally.
150 pidfile /var/run/redis_6379.pid
151
152 # Specify the server verbosity level.
```

p163 logfile 需要指定，利用端口号命名，放到redis根目录

```
# output for logging but
null
logfile "log6379.log"
```

save 900（秒） 1（变动的数据条数）

当900以内，至少有1条数据变动，flush保存数据到文件

save 300 10

300秒以内至少10条数据变动，保存文件

save 60 10000

```
198 # like in the following example:
199 #
200 # save ""
201
202 save 900 1
203 save 300 10
204 save 60 10000
205
```

p237左右，指定dump的持久化文件，每个服务单独指向一个文件

重启时，数据不会错乱

zero that will

# tell the loading code to :  
rdbchecksum yes

# The filename where to dump  
dbfilename dump6379.rdb

启动第二个和第三个redis实例

redis-server redis.conf (指定启动文件)

复制一份redis.conf，命名为redis6380.conf

编辑利用vim的替换命令

:%s/6379/6380/g 该命令为匹配到6379，将其替换为6380

需要第二个实例的配置文件

需要第三个实例的配置文件

拷贝redis.conf,用redis6380.conf,redis6381.conf

```
root      16942      1  0 11:23 ?          00:00:00 redis-server *:6379
root      16948 14903  0 11:23 pts/1    00:00:00 grep redis
[root@10-9-62-65 redis-3.2.11]# cp redis.conf redis6380.conf
[root@10-9-62-65 redis-3.2.11]# cp redis.conf redis6381.conf
[root@10-9-62-65 redis-3.2.11]#
```

将拷贝的文件中只修改与端口有关内容

port

```
# Accept connections on the specified port:
# If port 0 is specified Redis will use the default port
port 6380
```

pid文件

```
# Creating a pid file is best effort:
# nothing bad happens, the server will
# try to create the pid file, but it may fail.
pidfile /var/run/redis_6380.pid
```

6381的略

启动另外两个节点

```
#redis-server redis6380.conf
```

```
#redis-server redis6381.conf
```

```
#ps -ef|grep redis
```

```
[root@10-9-62-65 redis-3.2.11]# cp redis.conf redis6380.conf
[root@10-9-62-65 redis-3.2.11]# cp redis.conf redis6381.conf
[root@10-9-62-65 redis-3.2.11]# vim redis6380.conf
[root@10-9-62-65 redis-3.2.11]# vim redis6381.conf
[root@10-9-62-65 redis-3.2.11]# redis-server redis6380.conf
[root@10-9-62-65 redis-3.2.11]# redis-server redis6381.conf
[root@10-9-62-65 redis-3.2.11]# ps -ef|grep redis
root      16942      1  0 11:23 ?        00:00:00 redis-server *:6379
root      17022      1  0 11:29 ?        00:00:00 redis-server *:6380
root      17026      1  0 11:30 ?        00:00:00 redis-server *:6381
root      17036 14903  0 11:30 pts/1    00:00:00 grep  redis
[root@10-9-62-65 redis-3.2.11]#
```

指定端口登录客户端redis-cli -p [端口号] -h [ip]

```
#redis-cli -p 6380
```

```
Last login: Wed Jan 17 08:53:06 2018 from 60.247.62.254
[root@10-9-62-65 ~]# redis-cli -p 6380
127.0.0.1:6380>
```

```
#redis-cli -p 6381
```

```
Last login: Wed Jan 17 11:14:43 2018 from 60.247.62.254
[root@10-9-62-65 ~]# redis-cli -p 6381
127.0.0.1:6381>
```