

NSD OPERATION DAY06

- 1. [案例1：源码安装Redis缓存服务](#)
- 2. [案例2：常用Redis数据库操作指令](#)
- 3. [案例3：配置Redis主从服务器](#)

1 案例1：源码安装Redis缓存服务

1.1 问题

本案例要求先快速搭建好一台Redis服务器，并测试该缓存服务器：

- 设置变量test，值为123
- 查看变量test的值
- 设置计数器mycounter
- 对计数器mycounter进行增量加1操作

1.2 方案

使用源码包安装Redis服务器，使用redis-cli客户端工具连接Redis服务器并测试缓存数据库。
使用redis-cli命令测试Redis服务时可以使用的命令列表如表-1所示。

表 - 1 Redis命令列表

命令	作用
ping	测试服务器连接是否正确
set	设置变量并赋值
get	查看变量的值
incr	自增操作

1.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：安装Redis服务器

1) 源码安装Redis软件

```
01. [ root@svr5 ~] # tar - xzf redis- 3.0.6.tar.gz
02. [ root@svr5 ~] # cd redis- 3.0.6
03. [ root@svr5 ~] # make
04. [ root@svr5 ~] # make install
05. [ root@svr5 ~] # ./utils/install_server.sh           //初始化
06. Welcome to the redis service installer
07. This script will help you easily set up a running redis server
08.
09. Please select the redis port for this instance: [ 6379] //设置端口号，默认即可
```

[Top](#)

```

10.    Selecting default: 6379
11.    Please select the redis config file name [ /etc/redis/6379.conf ] //配置文件
12.    Selected default - /etc/redis/6379.conf
13.    Please select the redis log file name [ /var/log/redis_6379.log ] //日志文件
14.    Selected default - /var/log/redis_6379.log
15.    Please select the data directory for this instance [ /var/lib/redis/6379]
16.    //数据目录
17.    Selected default - /var/lib/redis/6379
18.    Please select the redis executable path [ /usr/local/bin/redis-server ]
19.    //Redis服务器软件存储路径
20.    Selected config:
21.    Port      : 6379
22.    Config file : /etc/redis/6379.conf
23.    Log file   : /var/log/redis_6379.log
24.    Data dir   : /var/lib/redis/6379
25.    Executable : /usr/local/bin/redis-server
26.    Cli Executable : /usr/local/bin/redis-cli
27.    Is this ok? Then press ENTER to go on or Ctrl-C to abort.
28.    //确认信息是否正确，回车确认即可
29.    Copied /tmp/6379.conf => /etc/init.d/redis_6379
30.    Installing service...
31.    Successfully added to chkconfig!
32.    Successfully added to runlevels 345!
33.    Starting Redis server...
34.    Installation successful!
35.
36.    [ root@svr5 ~ ] # ls /etc/init.d/redis_6379      //查看启动脚本

```

2) 启用Redis服务并查看监听端口状态

```

01.    [ root@svr5 ~ ] # /etc/init.d/redis_6379 restart
02.    [ root@svr5 ~ ] # netstat -nltlp | grep redis
03.    tcp      0      0 0.0.0.0:6379 0.0.0.0:* LISTEN    5749/redis-server *
04.    tcp      0      0 0:::6379 :::* LISTEN    5749/redis-server *

```

步骤二：测试缓存数据库

1) 使用redis-cli测试数据库

[Top](#)

```
01. [root@srv5 ~]# redis-cli
02. 127.0.0.1:6379> ping //测试服务器
03. PONG
04. 127.0.0.1:6379> set test 123 //设置变量
05. OK
06. 127.0.0.1:6379> get test //查看test值
07. "123"
08. 127.0.0.1:6379> INCR my counter //设置计数器my counter
09. (integer) 1
10. 127.0.0.1:6379> INCR my counter //对计数器my counter进行自增运算
11. (integer) 2
```

2 案例2：常用Redis数据库操作指令

2.1 问题

- 沿用练习一，通过redis-cli工具，对Redis数据库各数据类型进行增删改查等操作，要求如下：
- 分别对Strings、Hash表、List列表三种数据类型进行增删改查等常见操作
- 设置数据缓存时间
- 清空所有数据
- 对数据库操作

2.2 方案

使用redis-cli客户端工具连接Redis服务器并测试缓存数据库。

使用redis-cli命令测试Redis服务是可以使用的命令列表如表-2所示。

表 - 2 Redis命令列表

命令	作用
set key value [ex seconds]	设置 Strings 类型数据
setrange key offset value	从偏移量开始复写 key 的特定位的值
strlen key	统计字符串长度
append key value	字符存在则追加，不存在则创建 key 及 value
setbit key offset value	对 key 所存储字符串，设置或清除特定偏移量上的位(bit)
bitcount key [start] [end]	统计字符串中被设置为 1 的比特位数量
decr key	将 key 中的值减 1，key 不存在则先初始化为 0，再减 1
decrby key decrement	将 key 中的值，减去 decrement
get key	返回 key 所存储的字符串值 如果 key 不存在则返回特殊值 nil 如果 key 的值不是字符串，则返回错误，get 只能处理字符串
getrange key start end	返回字符串值中的子字符串，截取范围为 start 和 end 负数偏移量表述从末尾计数，-1 表示最后一个字符，-2 表示倒数第二个字符
incr key	将 key 的值加 1，如果 key 不存在，则初始为 0 后再加 1 主要应用为计数器
incrby key increment	将 key 的值增加 increment
incrbyfloat key increment	为 key 中所存储的值加上浮点数增量 increment
mget key [key...]	一次获取一个或多个 key 的值，空格分隔，<具有原子性>
mset key value [key value ...]	一次设置多个 key 及值，空格分隔，<具有原子性>
hset key field value	将 hash 表中 field 值设置为 value
hget key field	获取 hash 表中 field 的值
hmset key field value [field value...]	同时给 hash 表中的多个 field 赋值
hmget key field [field...]	返回 hash 表中多个 field 的值
hgetall key	返回 hash 表中所有 field 的值
hdel key field [field...]	删除 hash 表中多个 field 的值，不存在则忽略
hkeys key	返回 hash 表中所有 field 名称
hvals key	返回 hash 表中所有 field 的值
lpush key value [value...]	将一个或多个值 value 插入到列表 key 的表头 Key 不存在，则创建 key
lrange key start stop	从开始位置读取 key 的值到 stop 结束
lpop key	移除并返回列表头元素数据，key 不存在则返回 nil
lindex key index	返回列表中第 index 个值
lset key index value	将 key 中 index 位置的值修改为 value
rpush key value [value...]	将 value 插入到 key 的末尾
del key [key...]	删除一个或多个 key
expire key seconds	设置 key 的生存周期
persist key	设置 key 永不过期
ttl key	查看 key 的生存周期
keys 匹配	找符合匹配条件的 key，特殊符号用\屏蔽

2.3 步骤

实现此案例需要按照如下步骤进行。

[Top](#)

步骤一：常用Redis数据操作指令

1) 使用redis-cli测试数据库 (字符串常见操作指南)

```
01. [root@srv5 ~]# redis-cli
02. 127.0.0.1:6379> ping
03. PONG
04. 127.0.0.1:6379> set string1 "hello the word" //设置字符串变量
05. OK
06. 127.0.0.1:6379> get string1 //查看字符串变量
07. "hello the word"
08. 127.0.0.1:6379> set string2 "hello" ex 5 //设置字符串变量，并设置过期时间为5秒
09. OK
10. 127.0.0.1:6379> get string2 //查看字符串变量
11. "hello"
12. 127.0.0.1:6379> get string2 //字符串过期后，查看该值为空
13. (nil)
14. 127.0.0.1:6379> get string1
15. "hello the word"
16. 127.0.0.1:6379> set string1 hello nx //仅当string1不存在时，才执行set指令
17. (nil)
18. 127.0.0.1:6379> set string1 hello xx //仅当string1存在时，才执行set指令
19. OK
20. 127.0.0.1:6379> get string1 //查看修改后string1的值
21. "hello"
22. 127.0.0.1:6379> set string1 "hello the world" //修改string1的值
23. OK
24. 127.0.0.1:6379> get string1
25. "hello the world"
26. 127.0.0.1:6379> setrange string1 6 "Redis" //从第6个字符开始替换string1的值
27. (integer) 15
28. 127.0.0.1:6379> get string1
29. "hello Redisorld"
30. 127.0.0.1:6379> strlen string1 //计算string1的长度
31. (integer) 15
32. 127.0.0.1:6379> append string1 xxx //对string1进行追加操作
33. (integer) 18
34. 127.0.0.1:6379> get string1
35. "hello Redisorldxxx"
36. 127.0.0.1:6379> append string1 " xxx"
37. (integer) 22
38. 127.0.0.1:6379> get string1
39. "hello Redisorldxxx xxx"
```

[Top](#)

```

40.
41.
42. 127.0.0.1:6379> setbit string2 0 1 //按位设置string2的值，0位为1
43. (integer) 0
44. 127.0.0.1:6379> setbit string2 1 1 //按位设置string2的值，1位为1
45. (integer) 0
46. 127.0.0.1:6379> setbit string2 2 1
47. (integer) 0
48. 127.0.0.1:6379> setbit string2 3 0
49. (integer) 0
50. 127.0.0.1:6379> get string2 //不可以查看所有的值
51. "\xe0"
52. 127.0.0.1:6379> bitcount string2 //统计string2中1的个数
53. (integer) 3
54. 127.0.0.1:6379> getbit string2 0 //查看string2第0位的值
55. (integer) 1
56. 127.0.0.1:6379> getbit string2 1 //查看string2第1位的值
57. (integer) 1
58.
59.
60.
61. 127.0.0.1:6379> decr string3 //递减运算，初始值为0
62. (integer) - 1
63. 127.0.0.1:6379> decr string3
64. (integer) - 2
65. 127.0.0.1:6379> decr string3
66. (integer) - 3
67. 127.0.0.1:6379> decr string3
68. (integer) - 4
69. 127.0.0.1:6379> decr string3
70. (integer) - 5
71. 127.0.0.1:6379> set string4 10 //自定义变量初始值为10
72. OK
73. 127.0.0.1:6379> decr string4 //对自定义变量进行递减运算
74. (integer) 9
75. 127.0.0.1:6379> decr string4
76. (integer) 8
77. 127.0.0.1:6379> decr string4
78. (integer) 7
79. 127.0.0.1:6379> decrby string4 2 //对变量进行递减2运算
80. (integer) 5

```

```

81. 127.0.0.1:6379> decrby string4 2
82. (integer) 3
83. 127.0.0.1:6379> get string4
84. "3"
85.
86.
87.
88. 127.0.0.1:6379> set string5 "hello the world" //设置字符串变量
89. OK
90. 127.0.0.1:6379> getrange string5 0 4 //查看字串的第0至第4位
91. "hello"
92. 127.0.0.1:6379> getrange string5 - 3 - 1 //查看字串的倒数第3位至倒数第1位
93. "rld"
94. 127.0.0.1:6379> incr page //对变量进行递增运算，初始值为0
95. (integer) 1
96. 127.0.0.1:6379> incr page
97. (integer) 2
98. 127.0.0.1:6379> incr page
99. (integer) 3
100. 127.0.0.1:6379> incr page
101. (integer) 4
102. 127.0.0.1:6379> set string6 10 //设置字符串变量为10
103. OK
104. 127.0.0.1:6379> incr string6 //对变量进行递增运算
105. (integer) 11
106. 127.0.0.1:6379> incr string6
107. (integer) 12
108. 127.0.0.1:6379> incrby string6 2 //对变量进行递增2运算
109. (integer) 14
110. 127.0.0.1:6379> incrby string6 2
111. (integer) 16
112. 127.0.0.1:6379> incrby string6 2
113. (integer) 18
114. 127.0.0.1:6379> incrby string6 2
115. (integer) 20
116. 127.0.0.1:6379> set num 16.1 //设置浮点数变量
117. OK
118. 127.0.0.1:6379> incrby float num 1.1 //对浮点数进行递增1.1运算
119. "17.2"
120. 127.0.0.1:6379> incrby float num 1.1
121. "18.3"

```

```

122. 127.0.0.1:6379> incrby float num 1.1
123. "19.4"
124. 127.0.0.1:6379> incrby float num 1.1
125. "20.5"

```

2) Hash表常见操作指南

```

01. 127.0.0.1:6379> hset hkey google " www.g.cn"
02. //设置hash表hkey , google列的值为www.g.cn
03. ( integer) 1
04. 127.0.0.1:6379> hset hkey baidu " www.baidu.com"
05. //设置hash表hkey , baidu列的值为www.baidu.com
06. ( integer) 1
07. 127.0.0.1:6379> hget hkey google //查看hash表hkey中google列的值
08. "www.g.cn"
09. 127.0.0.1:6379> hget hkey baidu //查看hash表hkey中baidu列的值
10. "www.baidu.com"
11. 127.0.0.1:6379> hmset site google "www.g.cn" baidu "www.baidu.com"
12. OK
13. //一次性设置hash表site的多个列与值
14. 127.0.0.1:6379> hmget site google baidu
15. 1) "www.g.cn"
16. 2) "www.baidu.com"
17. //一次性查看hash表site的多个列值
18. 127.0.0.1:6379> hgetall site //查看site表中所有的列与值
19. 1) "google"
20. 2) "www.g.cn"
21. 3) "baidu"
22. 4) "www.baidu.com"
23. 127.0.0.1:6379> hdel site google //删除site表中google列
24. ( integer) 1
25. 127.0.0.1:6379> hgetall site //验证删除效果
26. 1) "baidu"
27. 2) "www.baidu.com"
28. 127.0.0.1:6379> hmset site google "www.g.cn" baidu "www.baidu.com" sina "www.sina.cn"
29. OK
30. 127.0.0.1:6379> hkeys site //查看site表的所有列
31. 1) "baidu"
32. 2) "google"
33. 3) "sina"

```

[Top](#)

34. 127.0.0.1:6379> hv als site //查看site表中所有列的值
35. 1) "www.baidu.com"
36. 2) "www.g.cn"
37. 3) "www.sina.com"
38. 127.0.0.1:6379> hmget site google baidu //一次性查看site表中的多个列值
39. 1) "www.g.cn"
40. 2) "www.baidu.com"

3)List列表常见操作指南

01. 127.0.0.1:6379> lpush list1 a b c //创建列表并赋值
02. (integer) 3
03. 127.0.0.1:6379> lpush list2 a //创建列表并赋值
04. (integer) 1
05. 127.0.0.1:6379> lpush list2 b //给列表追加新值
06. (integer) 2
07. 127.0.0.1:6379> lpush list2 c //给列表追加新值
08. (integer) 3
09. 127.0.0.1:6379> lrange list1 0 - 1
10. //查看列表list1中的所有值，从0位到最后1位
11. 1) "c"
12. 2) "b"
13. 3) "a"
14. 127.0.0.1:6379> lrange list1 1 2 //查看列表中第1和2位的值
15. 1) "b"
16. 2) "a"
17. 127.0.0.1:6379> lrange list1 1 - 1 //查看列表中第1至最后1位的值
18. 1) "b"
19. 127.0.0.1:6379> lrange list2 0 - 1 //查看所有的值
20. 1) "a"
21. 2) "c"
22. 3) "b"
23. 4) "a"
24. 127.0.0.1:6379> lrange list2 0 - 1
25. 1) "t"
26. 2) "a"
27. 3) "c"
28. 4) "b"
29. 5) "a"

```

30. 127.0.0.1:6379> lpop list2
31. //返回list2列表头元素数据，并将该值从列表中删除，key不存在则返回nil
32. "t"
33. 127.0.0.1:6379> lrange list2 0 - 1 //验证结果
34. 1) "a"
35. 2) "c"
36. 3) "b"
37. 4) "a"
38. 127.0.0.1:6379> lpop list2 //接续删除头部元素
39. "a"
40. 127.0.0.1:6379> lrange list2 0 - 1 //验证结果
41. 1) "c"
42. 2) "b"
43. 3) "a"
44. 127.0.0.1:6379> lpop list2
45. "c"
46. 127.0.0.1:6379> lrange list2 0 - 1
47. 1) "b"
48. 2) "a"
49. 127.0.0.1:6379> lrange list4 0 - 1 //查看全部数据如下
50. 1) "f"
51. 2) "e"
52. 3) "d"
53. 4) "a"
54. 5) "c"
55. 127.0.0.1:6379> lindex list4 0 //返回list4中第0个值
56. "f"
57. 127.0.0.1:6379> lindex list4 1 //返回list4中第1个值
58. "e"
59. 127.0.0.1:6379> lindex list4 - 1 //返回list4中最后1个值
60. "c"
61. 127.0.0.1:6379> lindex list4 - 2 //返回list4中倒数第2个值
62. "a"
63. 127.0.0.1:6379> lrange list4 0 - 1
64. 1) "f"
65. 2) "e"
66. 3) "d"
67. 4) "a"
68. 5) "c"
69. 127.0.0.1:6379> lset list4 0 test //给list4的第0为插入值，值为test
70. OK

```

[Top](#)

```

71. 127.0.0.1:6379> lrange list 0 - 1 //验证结果
72. 1) "test"
73. 2) "d"
74. 3) "a"

```

4)其他操作指南

```

01. 127.0.0.1:6379> set my key "hello" //定义字符串变量
02. OK
03. 127.0.0.1:6379> get my key //查看变量
04. "hello"
05. 127.0.0.1:6379> del my key //删除变量
06. (integer) 1
07. 127.0.0.1:6379> get my key //验证结果
08. (nil)
09. 127.0.0.1:6379> set my key "hello" //定义变量即值
10. OK
11. 127.0.0.1:6379> get my key //查看有值
12. "hello"
13. 127.0.0.1:6379> get my key
14. "hello"
15. 127.0.0.1:6379> expire my key 10 //定义过期时间
16. (integer) 1
17. 127.0.0.1:6379> get my key //10秒后查看，无值
18. (nil)
19. 127.0.0.1:6379> set my key "hello" //设置变量
20. OK
21. 127.0.0.1:6379> persist my key //重新定义过期时间为，永久有效
22. (integer) 1
23. 127.0.0.1:6379> get my key
24. "hello"
25. 127.0.0.1:6379> get my key
26. "hello"
27. 127.0.0.1:6379> ttl my key
28. (integer) - 1 //永不过期
29. 127.0.0.1:6379> expire my key 10 //定义过期时间
30. (integer) 1
31. 127.0.0.1:6379> ttl my key //查看过期时间
32. (integer) 9
33. 127.0.0.1:6379> ttl my key

```

[Top](#)

```
34. (integer) 8
35. 127.0.0.1:6379> ttl my key
36. (integer) 7
37. 127.0.0.1:6379> ttl my key
38. (integer) 6
39. 127.0.0.1:6379> ttl my key
40. (integer) 5
41. 127.0.0.1:6379> ttl my key
42. (integer) 4
43. 127.0.0.1:6379> ttl my key
44. (integer) 3
45. 127.0.0.1:6379> ttl my key
46. (integer) 3
47. 127.0.0.1:6379> ttl my key
48. (integer) 2
49. 127.0.0.1:6379> ttl my key
50. (integer) 1
51. 127.0.0.1:6379> ttl my key
52. (integer) -2 //已经过期
53. 127.0.0.1:6379> get my key //查看my key 的值已经为空
54. (nil)
55. 127.0.0.1:6379> set my key "hello"
56. OK
57. 127.0.0.1:6379> keys * //查看数据库下所有数据
58. 1) "string6"
59. 2) "list7"
60. 3) "my key "
61. 4) "string4"
62. 5) "db"
63. 6) "num"
64. 7) "result"
65. 8) "hkey "
66. 9) "string5"
67. 10) "string1"
68. 11) "bit1"
69. 12) "page"
70. 13) "bit2"
71. 14) "site"
72. 15) "string2"
73. 16) "list1"
74. 17) "string3"
```

```

75. 18) "list6"
76. 127.0.0.1:6379> key s li*
77. 1) "list7"
78. 2) "list1"
79. 3) "list6"
80. 127.0.0.1:6379> key s s*
81. 1) "string6"
82. 2) "string4"
83. 3) "string5"
84. 4) "string1"
85. 5) "site"
86. 6) "string2"
87. 7) "string3"
88. 127.0.0.1:6379> key s string[ 15] //查看string1或string5
89. 1) "string5"
90. 2) "string1"
91. 127.0.0.1:6379> key s string[ 0- 9] //查看string0值9的数据
92. 1) "string6"
93. 2) "string4"
94. 3) "string5"
95. 4) "string1"
96. 5) "string2"
97. 6) "string3"
98. 127.0.0.1:6379> key s ?it* //使用通配符所有数据
99. 1) "bit1"
100. 2) "bit2"
101. 127.0.0.1:6379> select 1 //进入1数据库，默认数据库为0
102. OK
103. 127.0.0.1:6379[ 1]> key s * //在新数据库中查看数据为空
104. ( empty list or set)
105. 127.0.0.1:6379[ 1]> set test "test" //在数据库1中创建变量
106. OK
107. 127.0.0.1:6379[ 1]> get test //查看变量的值
108. "test"
109. 127.0.0.1:6379[ 1]> select 2 //进入2数据库
110. OK
111. 127.0.0.1:6379[ 2]> key s * //查看所有数据
112. ( empty list or set)
113. 127.0.0.1:6379[ 2]> select 1
114. OK
115. 127.0.0.1:6379[ 1]> key s *

```

```
116. 1) "test"
117. 127.0.0.1:6379[ 1] > select 0
118. OK
119. 127.0.0.1:6379> keys *
120. 1) "string6"
121. 2) "list7"
122. 3) "my key "
123. 4) "string4"
124. 5) "db"
125. 6) "num"
126. 7) "result"
127. 8) "hkey "
128. 9) "string5"
129. 10) "string1"
130. 11) "bit1"
131. 12) "page"
132. 13) "bit2"
133. 14) "site"
134. 15) "string2"
135. 16) "list1"
136. 17) "string3"
137. 18) "list6"
138. 127.0.0.1:6379> flushall //清空所有数据
139. OK
140. 127.0.0.1:6379> keys * //验证结果
141. 127.0.0.1:6379[ 2] > select 0
142. OK
143. 127.0.0.1:6379> set my key "hello"
144. OK
145. 127.0.0.1:6379> keys *
146. 1) "my key "
147. 127.0.0.1:6379> select 1
148. OK
149. 127.0.0.1:6379[ 1] > keys *
150. ( empty list or set)
151. 127.0.0.1:6379[ 1] > select 0
152. OK
153. 127.0.0.1:6379> keys *
154. 1) "my key "
155. 127.0.0.1:6379> move my key 1 //将数据库0中的my key 变量移动至数据库1
156. ( integer) 1
```

[Top](#)

```

157. 127.0.0.1:6379> keys * //在数据库0中查看为空
158. (empty list or set)
159. 127.0.0.1:6379> select 1 //进入数据库1
160. OK
161. 127.0.0.1:6379[1]> keys * //查看所有数据库
162. 1) "my key"
163. 127.0.0.1:6379[1]> lpush cost 1 8 7 2 5 //创建列表cost
164. (integer) 5
165. 127.0.0.1:6379[1]> sort cost //对列表值进行排序
166. 1) "1"
167. 2) "2"
168. 3) "5"
169. 4) "7"
170. 5) "8"

```

3 案例3：配置Redis主从服务器

3.1 问题

本案例要求先快速搭建好两台Redis服务器，实现两台服务器之间自动数据同步，具体要求如下：

- 主服务器IP为192.168.2.100
- 从服务器IP为192.168.2.200
- 主服务器认证密码为redis123
- 测试主从数据是否正常通过

3.2 方案

通过修改Redis配置文件，实现两台服务器之间的自动主从同步功能，方案拓扑如图-1所示。

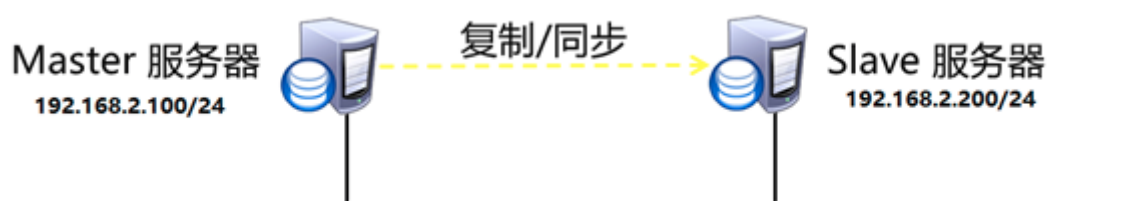


图-1

3.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：配置主从服务器设置

1) 主服务器安装Redis

[Top](#)

```
01 [root@svr100 ~]# tar -xzf redis-3.0.6.tar.gz
```

```

02. [ root@svr100 ~] # cd redis-3.0.6
03. [ root@svr100 ~] # make
04. [ root@svr100 ~] # make install
05. [ root@svr100 ~] # cd utils/
06. [ root@svr100 ~] # ./install_server.sh
07. Welcome to the redis service installer
08. This script will help you easily set up a running redis server
09.
10. Please select the redis port for this instance: [ 6379]
11. Selecting default: 6379
12. Please select the redis config file name [ /etc/redis/6379.conf ]
13. Selected default - /etc/redis/6379.conf
14. Please select the redis log file name [ /var/log/redis_6379.log]
15. Selected default - /var/log/redis_6379.log
16. Please select the data directory for this instance [ /var/lib/redis/6379]
17. Selected default - /var/lib/redis/6379
18. Please select the redis executable path [ /usr/local/bin/redis-server]
19. Selected config:
20. Port      : 6379
21. Config file : /etc/redis/6379.conf
22. Log file    : /var/log/redis_6379.log
23. Data dir    : /var/lib/redis/6379
24. Executable  : /usr/local/bin/redis-server
25. Cli Executable : /usr/local/bin/redis-cli
26. Is this ok? Then press ENTER to go on or Ctrl-C to abort.
27. Copied /tmp/6379.conf => /etc/init.d/redis_6379
28. Installing service...
29. Successfully added to chkconfig!
30. Successfully added to runlevels 345!
31. Starting Redis server...
32. Installation successful!

```

2) 从服务器安装Redis

```

01. [ root@svr200 ~] # tar -xzf redis-3.0.6.tar.gz
02. [ root@svr200 ~] # cd redis-3.0.6
03. [ root@svr200 ~] # make
04. [ root@svr200 ~] # make install
05. [ root@svr200 ~] # cd utils/
06. [ root@svr200 ~] # ./install_server.sh

```

[Top](#)


```

07. Welcome to the redis service installer
08. This script will help you easily set up a running redis server
09.
10. Please select the redis port for this instance: [ 6379]
11. Selecting default: 6379
12. Please select the redis config file name [ /etc/redis/6379.conf ]
13. Selected default - /etc/redis/6379.conf
14. Please select the redis log file name [ /var/log/redis_6379.log]
15. Selected default - /var/log/redis_6379.log
16. Please select the data directory for this instance [ /var/lib/redis/6379]
17. Selected default - /var/lib/redis/6379
18. Please select the redis executable path [ /usr/local/bin/redis-server]
19. Selected config:
20. Port      : 6379
21. Config file : /etc/redis/6379.conf
22. Log file   : /var/log/redis_6379.log
23. Data dir   : /var/lib/redis/6379
24. Executable : /usr/local/bin/redis-server
25. Cli Executable : /usr/local/bin/redis-cli
26. Is this ok? Then press ENTER to go on or Ctrl-C to abort.
27. Copied /tmp/6379.conf => /etc/init.d/redis_6379
28. Installing service...
29. Successfully added to chkconfig!
30. Successfully added to runlevels 345!
31. Starting Redis server...
32. Installation successful!

```

步骤二：配置主从服务器设置

1) 修改主服务器/etc/redis/6379.conf配置文件

```

01. [ root@svr100 ~] # vim /etc/redis/6379.conf
02. requirepass redis123 //设置服务器密码
03. [ root@svr100 ~] # /etc/init.d/redis_6379 restart //重启服务

```

2) 修改主服务器的启动脚本，添加服务器密码

```

01. [ root@svr100 ~] # vim /etc/init.d/redis_6379
02. $CLIXEC - a redis123 - p $REDISPORT shutdown

```

[Top](#)

3) 修改从服务器/etc/redis/6379.conf配置文件

01. [root@svr200 ~] # vim /etc/redis/6379.conf
02. slaveof 192.168.4.100 6379
03. masterauth redis123
04. [root@svr200 ~] # /etc/init.d/redis_6379 restart

步骤三：验证效果

1) 主服务器操作

01. [root@svr100 ~] # redis-cli -h 192.168.4.100 -a redis123 //登录主服务器设置数据
02. 192.168.4.10:6379> set test 123456
03. OK

2) 从服务器操作

01. [root@svr200 ~] # redis-cli -h 192.168.4.200 //登录主服务器查看数据同步效果
02. 192.168.4.20:6379> set test
03. “ 123456”