**Redis**

# Redis的安装

## Redis的安装

Redis是c语言开发的。

安装redis需要c语言的编译环境。如果没有gcc需要在线安装。

yum install gcc-c++

**yum -y install gcc automake autoconf libtool make**

安装步骤：

第一步：redis的源码包上传到linux系统。

第二步：解压缩redis。

第三步：编译。进入redis源码目录。make

第四步：安装。make install PREFIX=/usr/local/redis

PREFIX参数指定redis的安装目录。一般软件安装到/usr/local/redis目录下

## 连接redis

### redis的启动：

前端启动：在redis的安装目录下直接启动redis-server

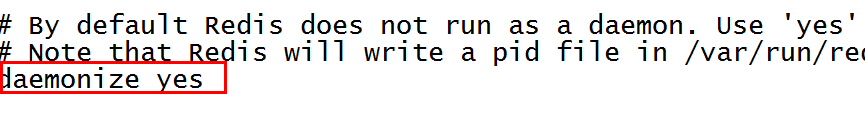
[root@localhost bin]# ./redis-server

后台启动：

把/root/redis-3.0.0/redis.conf复制到/usr/local/redis/bin目录下

[root@localhost redis-3.0.0]# cp redis.conf /usr/local/redis/bin/

修改配置文件：



[root@localhost bin]# ./redis-server redis.conf

查看redis进程：

[root@localhost bin]# ps aux|grep redis

root 5190 0.1 0.3 33936 1712 ? Ssl 18:23 0:00 ./redis-server \*:6379

root 5196 0.0 0.1 4356 728 pts/0 S+ 18:24 0:00 grep redis

[root@localhost bin]#

### Redis-cli (客户端)

[root@localhost bin]# ./redis-cli

默认连接localhost运行在6379端口的redis服务。

[root@localhost bin]# ./redis-cli -h 192.168.18.133 -p 6379

-h：连接的服务器的地址

-p：服务的端口号

关闭redis：[root@localhost bin]# ./redis-cli shutdown

# Redis五种数据类型

String：key-value（做缓存）

Redis中所有的数据都是字符串。命令不区分大小写，key是区分大小写的。Redis是单线程的。Redis中不适合保存内容大的数据。Redis的key都是字符串，对于值来讲，有五种数据类型。分别为：String、hash、Set、List、sortedSet

get、set、

incr：加一（生成id）

Decr：减一

Hash：key-fields-values（做缓存）

相当于一个key对于一个map，map中还有key-value

使用hash对key进行归类。

Hset：向hash中添加内容

Hget：从hash中取内容

打开redis后台服务

[root@redis1 bin]# ./redis-server redis.conf

使用客户端连接redis

[root@redis1 bin]# ./redis-cli -p 192.168.18.134 -p 6379

**2.1字符串**

127.0.0.1:6379> set name zhangsan

OK

127.0.0.1:6379> set age 22

OK

127.0.0.1:6379> set address beijing

OK

127.0.0.1:6379> set idcard 110210199101202413

OK

127.0.0.1:6379> keys \*

1) "age"

2) "address"

3) "name"

4) "index"

5) "idcard"

127.0.0.1:6379> get age

"22"

127.0.0.1:6379> get address

"beijing"

127.0.0.1:6379>

**自增：**

127.0.0.1:6379> incr currentPage

(integer) 1

127.0.0.1:6379> incr currentPage

(integer) 2

127.0.0.1:6379> incr currentPage

(integer) 3

127.0.0.1:6379> keys \*

1) "age"

2) "currentPage"

3) "address"

4) "name"

5) "index"

6) "idcard"

127.0.0.1:6379> get currentPage

"3"

**自减：**

127.0.0.1:6379> decr currentPage

(integer) 2

127.0.0.1:6379> decr currentPage

(integer) 1

127.0.0.1:6379> decr currentPage

(integer) 0

127.0.0.1:6379> decr currentPage

(integer) -1

127.0.0.1:6379> decr currentPage

(integer) -2

127.0.0.1:6379> get currentPage

"-2"

**删除：del**

127.0.0.1:6379> keys \*

1) "age"

2) "currentPage"

3) "address"

4) "name"

5) "index"

6) "idcard"

127.0.0.1:6379> del index

(integer) 1

127.0.0.1:6379> keys \*

1) "age"

2) "currentPage"

3) "address"

4) "name"

5) "idcard"

**2.2 Hash类型：**

127.0.0.1:6379> hset persons name lisi

(integer) 1

127.0.0.1:6379> hget persons name

"lisi"

127.0.0.1:6379> hset persons age 20

(integer) 1

127.0.0.1:6379> hset persons address changsha

(integer) 1

127.0.0.1:6379> hset persons birthday 1997-10-10

(integer) 1

127.0.0.1:6379> keys \*

1) "persons"

2) "age"

3) "currentPage"

4) "address"

5) "name"

6) "idcard"

127.0.0.1:6379> hkeys persons

1) "name"

2) "age"

3) "address"

4) "birthday"

127.0.0.1:6379> hvals persons

1) "lisi"

2) "20"

3) "changsha"

4) "1997-10-10"

127.0.0.1:6379> hgetall persons

1) "name"

2) "lisi"

3) "age"

4) "20"

5) "address"

6) "changsha"

7) "birthday"

8) "1997-10-10"

**删除:**

127.0.0.1:6379> hdel persons age

(integer) 1

127.0.0.1:6379> hgetall persons

1) "name"

2) "lisi"

3) "address"

4) "changsha"

5) "birthday"

6) "1997-10-10"

**2.3 List类型：**

**向左添加**:

127.0.0.1:6379> lpush ages 1 2 3 4 5 6 7 8 9

(integer) 9

127.0.0.1:6379> rpush names zhangsan lisi wangwu zhaoliu tianqi

(integer) 5

127.0.0.1:6379> lrange ages 0 -1

1) "9"

2) "8"

3) "7"

4) "6"

5) "5"

6) "4"

7) "3"

8) "2"

9) "1"

**向右添加:**

127.0.0.1:6379> rpush names zhangsan lisi wangwu zhaoliu tianqi

127.0.0.1:6379> lrange ages 0 -1

1) "9"

2) "8"

3) "7"

4) "6"

5) "5"

6) "4"

7) "3"

8) "2"

9) "1"

**从左边取值**：  
127.0.0.1:6379> lpop ages

"9"

127.0.0.1:6379> lpop ages

"8"

127.0.0.1:6379>

127.0.0.1:6379> lrange ages 0 -1

1) "7"

2) "6"

3) "5"

4) "4"

5) "3"

6) "2"

7) "1"

127.0.0.1:6379>

127.0.0.1:6379> lrange ages 0 -1

1) "7"

2) "6"

3) "5"

4) "4"

5) "3"

6) "2"

127.0.0.1:6379> lrange ages 0 -1

1) "7"

2) "6"

3) "5"

4) "4"

5) "3"

6) "2"

**2.4 Set集合类型：**

192.168.18.134:6379> sadd empNames zhangsan lisi wangwu zhaoliu tianqi

(integer) 5

192.168.18.134:6379> sadd empNames lisi

(integer) 0

192.168.18.134:6379> srem empNames zhaoliu

(integer) 1

192.168.18.134:6379> smembers empNames

1) "zhangsan"

2) "lisi"

3) "wangwu"

4) "tianqi"

192.168.18.134:6379> sadd seta a b c d e

(integer) 5

192.168.18.134:6379> sadd setb c d e f g

(integer) 5

192.168.18.134:6379> sdiff seta setb

1) "a"

2) "b"

192.168.18.134:6379> sdiff setb seta

1) "f"

2) "g"

192.168.18.134:6379> sinter seta setb

1) "d"

2) "c"

3) "e"

192.168.18.134:6379> sunion seta setb

1) "d"

2) "a"

3) "b"

4) "c"

5) "e"

6) "f"

7) "g"

**2.5 SortedSet类型**

**1**92.168.18.134:6379> zadd students 1 zhangsan 3 lisi 2 wangwu 5 zhaoliu

(integer) 4

192.168.18.134:6379> zrange students 0 -1

1) "zhangsan"

2) "wangwu"

3) "lisi"

4) "zhaoliu"

192.168.18.134:6379> zrem students lisi

(integer) 1

192.168.18.134:6379> zrange students 0 -1

1) "zhangsan"

2) "wangwu"

3) "zhaoliu"

192.168.18.134:6379> zrevrange students 0 -1

1) "zhaoliu"

2) "wangwu"

3) "zhangsan"

192.168.18.134:6379> zrevrange students 0 -1 withscores

1) "zhaoliu"

2) "5"

3) "wangwu"

4) "2"

5) "zhangsan"

6) "1"

设置key的过期时间:

设置name的过期时间为100秒

192.168.18.134:6379> expire name 100

使用ttl name可以查看key的剩余时间

192.168.18.134:6379> ttl name

(integer) 73

192.168.18.134:6379> ttl name

192.168.18.134:6379> ttl name

(integer) 59

192.168.18.134:6379> ttl name

(integer) 57

192.168.18.134:6379> ttl name

(integer) -2

-1表示持久化

-2表示过期

192.168.18.134:6379> set name zhangsan

OK

192.168.18.134:6379> expire name 90

(integer) 1

192.168.18.134:6379> ttl name

(integer) 86

192.168.18.134:6379> expire name 90 //如果对某个key设置了过期时间，在倒计时的过程中要重置过期时间可以重新设置

(integer) 1

192.168.18.134:6379> ttl name

(integer) 88

192.168.18.134:6379> persist name //key设置了过期时间，想让保留可以使用persist 命令

(integer) 1

192.168.18.134:6379> ttl name

(integer) -1

# Redis的持久化方案

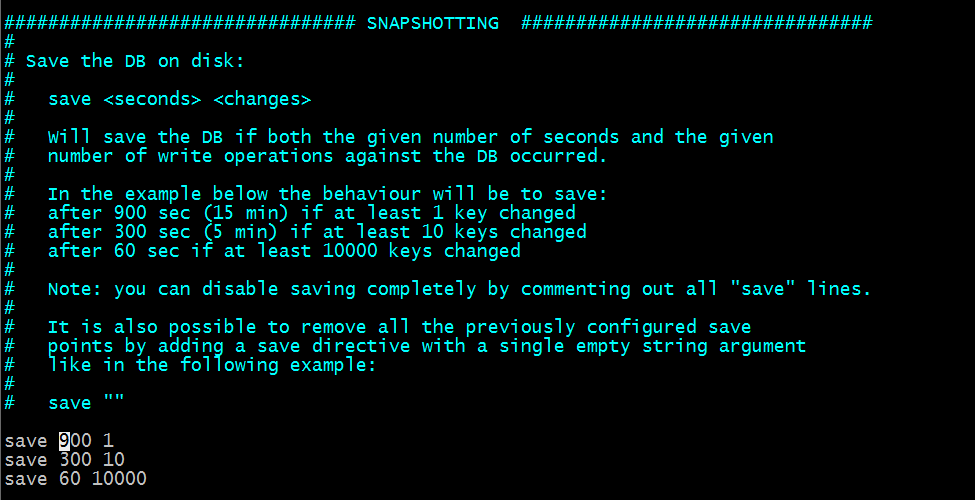
Redis的所有数据都是保存到内存中的。

Rdb：快照形式，定期把内存中当前时刻的数据保存到磁盘。Redis默认支持的持久化方案。

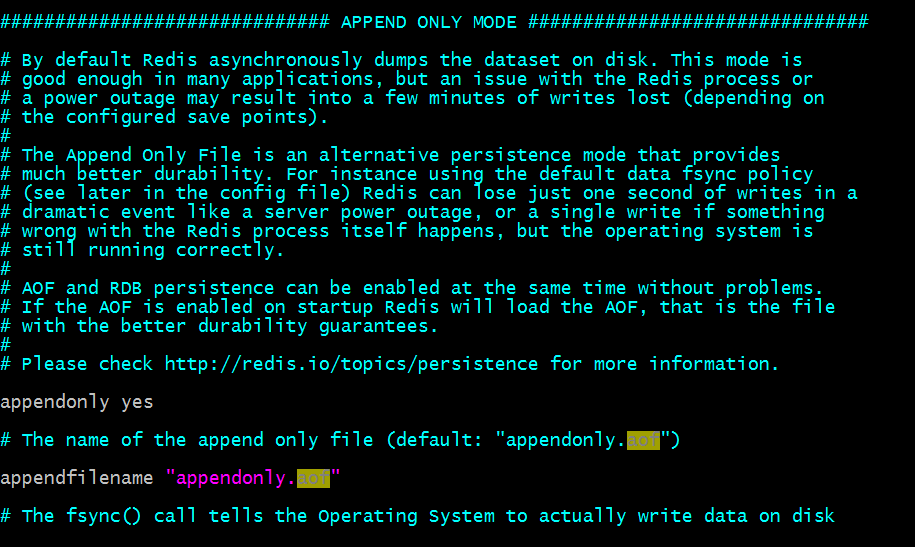
aof形式：append only file。把所有对redis数据库操作的命令，增删改操作的命令。保存到文件中。数据库恢复时把所有的命令执行一遍即可。

在redis.conf配置文件中配置。

Rdb：



Aof的配置：



两种持久化方案同时开启使用aof文件来恢复数据库。

# Redis集群

**1、使用ruby脚本搭建集群。需要ruby的运行环境。**

安装ruby

yum install ruby

yum install rubygems

**2.安装ruby脚本运行使用的包**

把redis-3.0.0/src/下的redis-trib.rb拷贝到/usr/local/redis-cluster

[root@redis1 src]# cp -f redis-trib.rb /usr/local/redis-cluster

[root@localhost ~]# gem install redis-3.0.0.gem

3、把redis.conf中的#cluster-enabled yes，把#号去掉

4、在/usr/local 下建一个redis-cluster目录,把redis目录复制到redis-cluster目录中,拷贝6份，分别取名为redis01、redis02、redis03、redis04、redis05、redis06

5、把每个redis中的redis.conf中的port分别改为7001,7002,7003,7004,7005,7006

在redis-cluster目录下新一个start-all.sh文件

内容如下:

cd redis01/bin

./redis-server redis.conf

cd ../..

cd redis02/bin

./redis-server redis.conf

cd ../..

cd redis03/bin

./redis-server redis.conf

cd ../..

cd redis04/bin

./redis-server redis.conf

cd ../..

cd redis05/bin

./redis-server redis.conf

cd ../..

cd redis06/bin

./redis-server redis.conf

cd ../..

**6、增加start-all.sh文件的执行权限**

Chmod u+x start-all.sh

**7、运行start-all.sh**

[root@redis1 redis-cluster]# ./start-all.sh

[root@redis1 redis-cluster]# ps aux | grep redis

root 1202 0.1 0.7 137456 7496 ? Ssl 07:54 0:00 ./redis-server \*:7001 [cluster]

root 1204 0.1 0.7 137456 7496 ? Ssl 07:54 0:00 ./redis-server \*:7002 [cluster]

root 1208 0.1 0.7 137456 7496 ? Ssl 07:54 0:00 ./redis-server \*:7003 [cluster]

root 1212 0.1 0.7 137456 7496 ? Ssl 07:54 0:00 ./redis-server \*:7004 [cluster]

root 1216 0.1 0.7 137456 7496 ? Ssl 07:54 0:00 ./redis-server \*:7005 [cluster]

root 1220 0.1 0.7 137456 7496 ? Ssl 07:54 0:00 ./redis-server \*:7006 [cluster]

root 1226 0.0 0.0 103256 876 pts/0 S+ 07:54 0:00 grep redis

**8、使用ruby脚本搭建集群**

./redis-trib.rb create --replicas 1 192.168.18.132:7001 192.168.18.132:7002 192.168.18.132:7003 192.168.18.132:7004 192.168.18.132:7005 192.168.18.132:7006

**9、创建关闭集群的脚本：**

[root@localhost redis-cluster]# vim stop-all.sh

redis01/bin/redis-cli -p 7001 shutdown

redis02/bin/redis-cli -p 7002 shutdown

redis03/bin/redis-cli -p 7003 shutdown

redis04/bin/redis-cli -p 7004 shutdown

redis05/bin/redis-cli -p 7005 shutdown

redis06/bin/redis-cli -p 7006 shutdown

# 集群的使用方法

Redis-cli连接集群。

[root@localhost redis-cluster]# redis01/bin/redis-cli -p 7001 -c

-c：代表连接的是redis集群

# Jedis

需要把jedis依赖的jar包添加到工程中。Maven工程中需要把jedis的坐标添加到依赖。

推荐添加到服务层。shop-content-service工程中。