# redis-cli info | grep version

# redis-cli dbsize 查看redis的key的数量

# redis-cli -n 3 dbsize 查看记录数

# redis-cli info 查看redis信息

**# redis-cli info | grep used\_memory\_human 查看redis占用的内存**

# redis-cli info memory 查看redis占用的内存详细

# redis-cli info keyspace 查看redis的key的数量

# redis-cli info | grep ^db 列出redis的数据库

# redis-cli scan 0 列出db0全部key , scan要2.8以上版本可用

# redis-cli -n 0 keys '\*'

# redis-cli keys '\*' 列出db0全部key

# redis-cli keys 'a\*' 等同redis-cli --scan --pattern '\*a\*'

**key相关操作**

# redis-cli keys "\*"

# redis-cli keys "\*" | while read line; do echo $line;done

# redis-cli keys "\*" | while read line;do ttl=`redis-cli ttl $line`;echo $line $ttl;done 打印每个key及其ttl

# redis-cli keys "\*" | while read line;do ttl=`redis-cli ttl $line`;[ $ttl -ge 3600 ] && echo $line $ttl; done 打印ttl大于3600的key

# redis-cli keys "\*" | while read line;do ttl=`redis-cli ttl $line`;if [ $ttl -ge 3600 ]; then redis-cli del $line; fi; done 删除ttl大于3600的key

# redis-cli keys 'index\*' 列出index开头的key

# redis-cli keys 'a\*' | xargs -ti redis-cli get {} 列出index开头的key的值

# redis-cli keys 'a\*' | xargs -ti redis-cli del {}

# redis-cli keys 'a\*' |xargs redis-cli del 删除a开头的key

# redis-cli -n 1 keys 'a\*' | xargs redis-cli -n 1 del 对redis下的指定库进行操作 需要加 -n 库序号

# redis-cli del abc 删除key为abc的值

# redis-cli set test "hello abc"

# redis-cli get test

# redis-cli -a 密码 flushdb 删除当前数据库中的所有Key

# redis-cli -a 密码 flushall 删除所有数据库中的key

**redis-benchmark为Redis性能测试工具**

# redis-benchmark

# redis-benchmark -t set -n 100000 -r 10000000 测试set速度

# redis-benchmark -h localhost -p 6379 -c 100 -n 100000 100个并发连接，100000个请求，检测host为localhost 端口为6379的redis服务器性能

**redis-check-aof: 用于日志检查和恢复  
redis-check-dump：用于redis数据文件检查**

# redis-check-dump  dump.rdb 检查本地数据文件  
# redis-check-aof appendonly.aof 更新日志检查 ，加--fix参数为修复log文件

**Redis日志文件 /var/log/redis/redis.log**

redis服务关闭后，缓存数据会自动dump到硬盘上，硬盘地址为redis.conf中的配置项dir 和dbfilename dump.rdb所设定

与memcached一样，为了保证效率，数据都是缓存在内存中。区别的是redis会周期性的把更新的数据写入磁盘或者把修改操作写入追加的记录文件

**# redis-cli 进入redis命令行模式**

> info 服务器基本信息

**> help del 查看del的帮助**

> set say 12313123 设置say的值为

> get say

> setnx job bbbbbbbbb

> **exists aaa** 当前的key是否存在

**> expire aaa 15 设置aaa的过期时间为15秒**

**> del aaa 删除key为aaa的值**

**> select 2 数据库切换**

**> dbsize 返回当前数据库中key的数目**

****> config get max\*****

**> config get dir 获取服务器的参数dir的配置**

**> config get maxclients 查看最大连接数**  运行过程中调整参数config set maxclients 10001

**> client list** 列出所有客户端连接

> client kill 127.0.0.1:11902 关闭ip:port的客户端

**> SLOWLOG get**

**> flushdb 清空当前数据库**

**> flushall 清除所有数据库**

**> bgsave** 在后台异步保存当前**数据库的数据到磁盘**

**> save 同步保存当前数据库的数据到磁盘**

**> lastsave 返回最近一次 Redis 成功执行保存操作的时间点( [SAVE](http://www.cnblogs.com/ikodota/archive/2012/03/05/php_redis_cn.html" \l "svr_SAVE) 、 [BGSAVE](http://www.cnblogs.com/ikodota/archive/2012/03/05/php_redis_cn.html" \l "svr_BGSAVE) 等)，以 UNIX 时间戳格式表示**

**> quit**

# redis-server /etc/redis.conf 启动redis

# redis-cli shutdown   关闭redis

# redis-cli -p 6397 shutdown   关闭6397端口的redis

# redis-cli save 强制备份数据到磁盘 或者redis-cli -p 6380 save（指定端口）

# redis-cli -h localhost -p 6379 monitor  监控host为localhost，端口为6379 edis的连接及读写操作   
# redis-cli -h localhost -p 6379 info  提供host为localhost，端口为6379 redis服务的统计信息

setnx，判断值是否存在，如果存在就不更新键值。如果不存在，则赋值。 （nx ：not exist）

getset，设置一个新值，并返回key的旧值。

mset，批量设置键值。

2. 输入密码有两种方式：

输入任何命令前，在客户端界面执行：auth 123456

进入客户端时输入密码：redis-cli -a 123456

**redis开启远程连接**

修改redis.conf

bind 127.0.0.1为bind 0.0.0.0

**Redis手册**

**http://doc.redisfans.com/string/getset.html**

**http://www.cnblogs.com/ikodota/archive/2012/03/05/php\_redis\_cn.html**

****http://redisdoc.com/index.html****

**Redis应用场景**

**www.linuxidc.com/Linux/2013-09/90120p3.htm**

**[redis实现有序的消息队列](http://www.cnblogs.com/haython/p/3529112.html)**

**http://www.cnblogs.com/haython/p/3529112.html**

**找出拖慢 Redis 的罪魁祸首**，要想知道在 Redis 实例内部都做了些什么是非常困难的。幸运的是 Redis 提供了一个下面这样的命令统计工具：

**> INFO commandstats**

**redis数据自动持久化规则配置**save 900 1  
save 300 10  
save 60 10000

**# redis-cli info | grep memory\_human**

used\_memory:13490096 数据占用了多少内存（字节）

used\_memory\_human:12.87M 数据占用了多少内存（带单位的，可读性好）

used\_memory\_rss:13490096   redis占用了多少内存

used\_memory\_peak:15301192 占用内存的峰值（字节）

used\_memory\_peak\_human:14.59M  占用内存的峰值（带单位的，可读性好）

used\_memory\_lua:31744 lua引擎所占用的内存大小（字节）