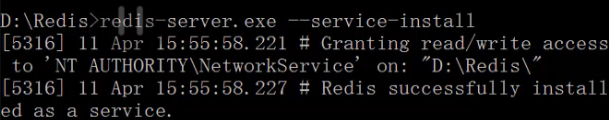
1：下载https://github.com/MicrosoftArchive/redis/releases 压缩包



解压后打开cmd

2:输入redis-server.exe --service-install



打开服务，找到Redis，开启（因为没有改名字，所以系统会默认命名为Redis）



3：进入到Redis中去： Redis-cli.exe -h localhost -p 6379(连接到）





4：输入keys \* （查看有多少数据）

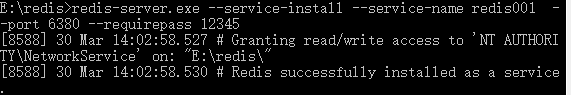
5：卸载 Redis：

先停掉服务后输入命令redis-server.exe --service-uninstall --service-name Redis（服务的名字）



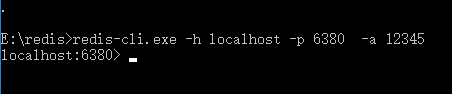
6：重新安装并设置参数：

redis-server.exe --service-install --service-name redis001 --port 6380 --requirepass 12345

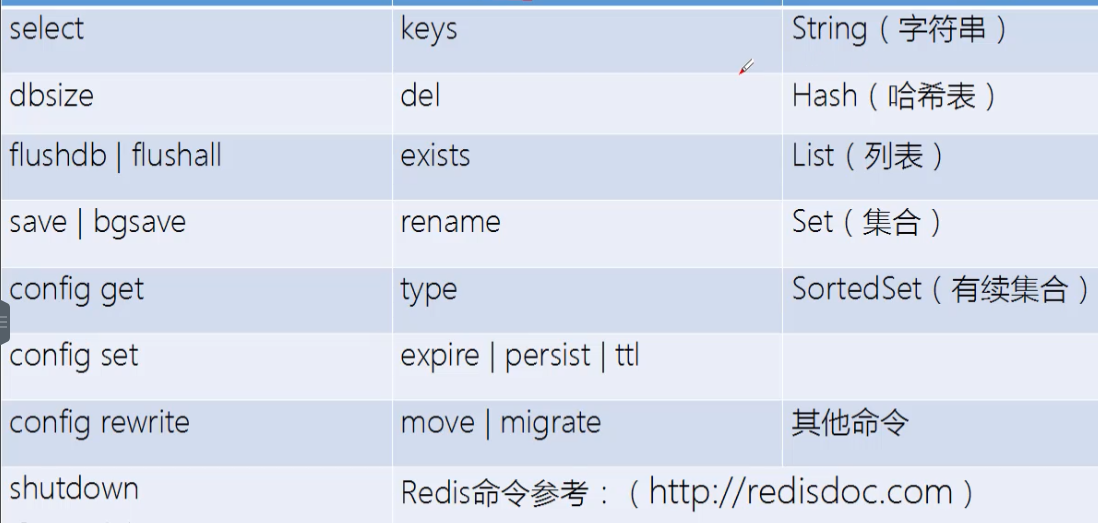




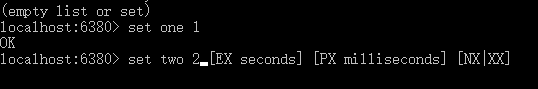
7：登录redis redis-cli.exe -h localhost -p 6380 -a 12345 （-p是端口号，-a 是密码 如果输错了密码，会进入，但是操作的时候回提示没有授权，继续输入：auth 正确密码就可以正常进入）



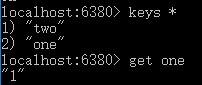
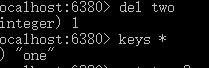
8redi常用命令：

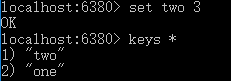


9：插入数据 set one key(后面的不用管，回车后悔消掉）



查看，对字符串操作

1 2

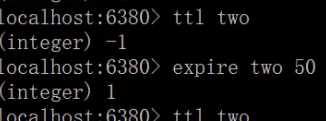
3set 这个命令，如果原来中key存在就修改，没有就添加

Exists key ,存在就返回1 ，不存在就返回0

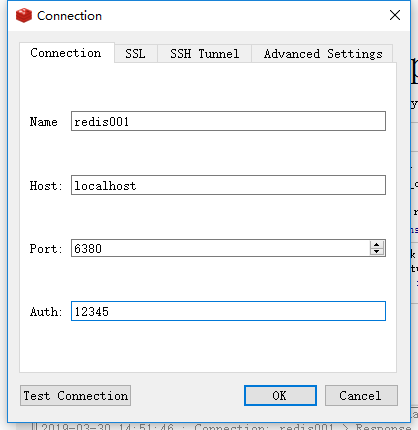
Rename one On 给key 改名

Type key 返回key 的数据类型。

Expire| persist | ttl expire key 时间 设置这个数据存活的时间| 取消key 上的存活时间限制，永生， |ttl key 看这个key 还可以活多久 返回的是 -1 代表永生，， 1 代表操作了1 条数据。



Redis 桌面管理工具 ---redis desktop manager 安装好后连接



Redis 一个数据库实例里面安装了16 个库（分16 部分进行存储）

Move redis下的16 个库中移动 migrate 可以在不同数据库中移动

Dbsize 查看当前库中有多少数据

Flushdb /flushall 清空当前库/清空所有库

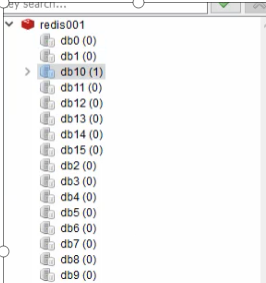
Save / bgsave save 由主进程保存，bgsave 创建一个子进程保存。

Config set/get 更改的属性 （改的值）

更改的属性只是当前有效，要想永久有效，需要写进去。用命令： config rewrite

Shutdown 关闭连接。

<http://redisdoc.com> 更多命令链接。



Redis 数据类型和 jedis API

Jedis JedisHash JedisList JedisSet JedisSortSet JedisString

使用Pipeline 存数据快，数据越多越快。

pipe.sync(); //将封装后的PIPE一次性发给redis

pipeline是多条命令的组合，为了保证它的原子性，redis提供了简单的事务

Redis 数据库的主要瓶颈是网络速度，其次是内存与cpu的速度和容量。

Pipeline 实现了批量发出请求/一次性获取响应，不必为每个请求都阻塞等待响应，提高了访问效率（吞吐量）

原理：批量发出请求，一次性获得响应 。

Redis的事物处理

ACID

原子性：要么全部成功，要么全部失败。

一致性：事物数据一致

隔离性：多个事物之间不能互相影响

永久性：当事物被提交后，这种修改是永久的

处理流程

Jedis.watch(key1,key2...)//监视key (从开始监视到结束只能被事物修改，其他类型的修改就会返回空)

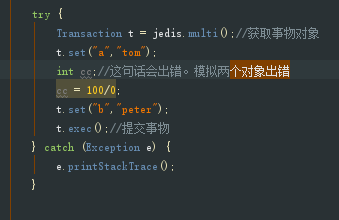
Trasction t = jedis.muti();//开始事物（在事物体当代码出错或者被本事物监视的事物被修改的，事物不会被提交）

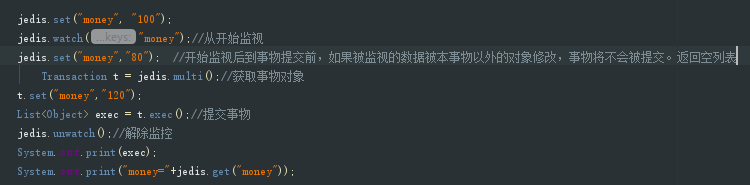
事物体（some，methods）

List<Obiect> r =t.exec()//事物提交

R.isEmpty();//判断事物是否为空

Jedis.unwatch（）//关闭事物。



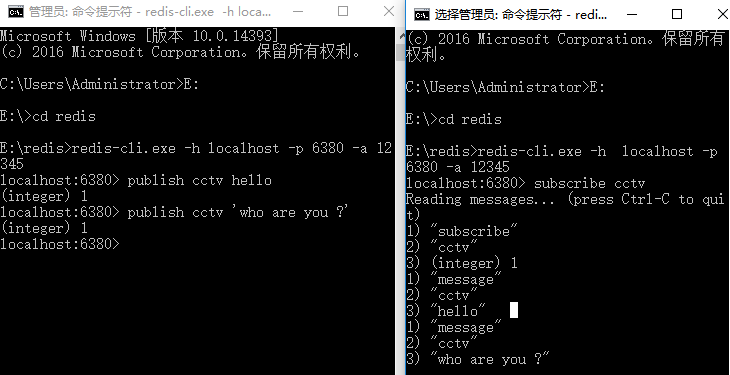


Redis消息通信

开两个cmd ,建立两个客户端

subscribe cctv 建立频道。

publish cctv hello 向CCTV 发hello 消息。



Redis 主从复制：

备份容错

一主多从

自动同步

主从库的配置：

（1） 配置文件： redis。Windows。Conf

slaveof <masterip> <masterport> masterip 主redis 的地址 masterport 端口号

masterauth <master-password> 主redis 的密码（password）

Slave-read-only yes (不用改) 从库只读

1. slaveof 命令。

Slaveof host port 将当前库设置为host、port 对应实例的从库

Slaveof no one 取消当前库的从库身份（已同步的数据保留）

一：配置文件安装从数据库

启动redisB

redis-server.exe --service-install redis.windows.conf --service-name redis002 --port 6381

已配置文件配置启动。

如果配置错误用命令

 sc delete 服务的名字 可以删除服务。



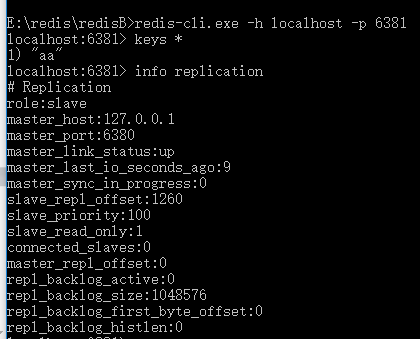
启动redis001（主） 添加数据。

Redis-cli.exe -h localhost -p 6380 -a 12345

查看重数据库数据

Redis-cli.exe -h localhost -p 6381

查看主从情况 ： info replication (命令都一样，显示不同)



二:命令方式安装从数据库

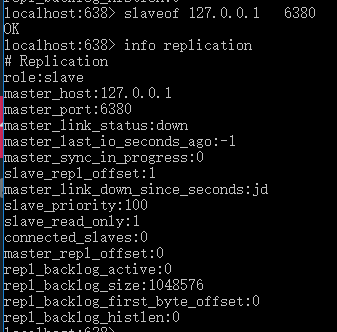
安装redis003（如果遇到系统提醒改软件无法再电脑运行的话，重新解压一份放在另外的文件夹里）



启动 Redis003 Redis-cli.exe -h localhost -p 638

（注意这个port 要和创建的一样，不然你的cmd 就会卡着不动）

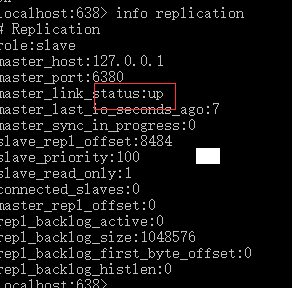
1：slaveof 127.0.0.1 6380 (通过info replication )来查看连接状态。



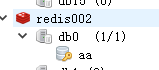
设置密码 config set masterauth 12345



Info replication 查看连接状态：



主从复制：（通过redis 的可视化工具查看）



哨兵（sentinel）：当master 挂掉后，通过合适 的算法选出一个合适的从数据库作为主数据库。

复制一个redis 再里面新建一个sentinel.conf 文件，下面温配置信息

port 26386 端口号

protected-mode no 关闭保护模式（默认）

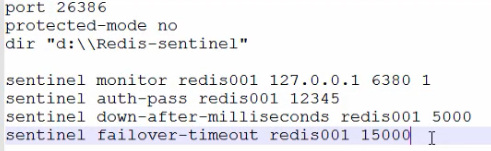
dir "d:\\redis\\redis\_sentinel" 哨兵路径

sentinel monitor redis001 127.0.0.1 6380 2 监控那个节点 （2 代表几个哨兵）

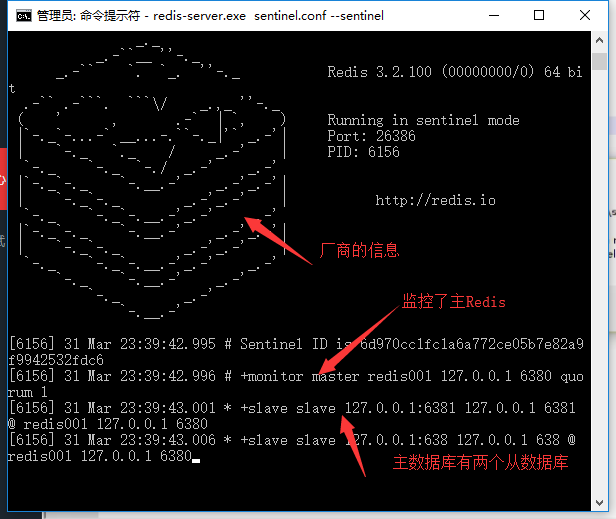
sentinel auth-pass redis001 12345 告诉监控对象的密码

sentinel down-after-milliseconds redis001 5000 时间间隔（多长时间没有反应就认为下线了）

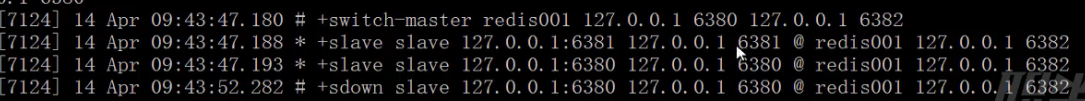
sentinel failover-timeout redis001 15000 故障迁移的超时时间



运行命令 redis-server.exe sentinel.conf --sentinel 出现如下如



关掉Redis001 ：如下信息



哨兵会选一个从节点，作为master ，如果 Redis001 开启，不会被系统认为master，作为一个从节点存在于系统中。

哨兵认为：主从节点的密码是相同的。

配置多个哨兵：

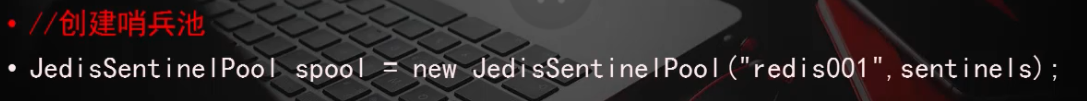
复制sentinel.conf 注意名字不要相同。全部启动

书写Java程序测试Redis（）

哨兵池 用SET<String> a = new HashSet《String》（）；创建哨兵集合

添加哨兵 a.add(“哨兵地址”)；

创建哨兵池 jedisSentinePool



从哨兵池里面获得连接对象



操作

关闭连接

当哨兵启动异常的时候，检测配置文件信息与实际的主从情况。（info replication），必要时从新配置文件。