**vim redis.conf**

daemonize yes //使redis能在后台运行，启用守护进程

pidfile /data/redisdata/6379/redis\_6379.pid //指定pid文件保存路径

loglevel debug //日志级别，学习阶段设为debug

bind 127.0.0.1 //绑定主机地址，未绑定端口

logfile /data/redisdata/6379/redis.log //指定日志文件目录

dir /data/redisdata/6379/dbdata/ //指定数据持久化RDB数据文件和AOF日志文件存放路径

logfile stdout // 日志记录方式，默认为标准输出，可自定义文件路径

appendonly yes //启动数据持久化AOF日志文件

timeout 300 // 指定redis客户端在多长时间后退出登录，如果是0则表示关闭该功能

appendfsync everysec //每秒钟执行一次AOF同步（默认）

no-appendfsync-on-rewrite no //启用AOF同步重写，当AOF日志文件过大时执行重写（默认）

auto-aof-rewrite-min-size 1Gb //当AOF文件超过1GB的100%时即2GB时自动启动日志文件重写（默认）//允许重写的最小AOF日志文件

maxmemory 3GB //配置最大内存（根据本机物理内存考量）

maxmemory-policy volatile-lfu //内存不足的缓存淘汰算法，我选的是——从所有配置了过期时间的键中驱逐使用频率最少的键

maxmemory-samples 5 //从淘汰算法选出的列表中随机选择的样本数量

requirepass "密码" //设置连接密码，客户端连接Redis时需要通过auth <password>命令提供密码（默认关闭，可选不设）

————————————————

redis数据持久化

数据持久化一共三种办法：

一、数据自动写入

1.appendfsync no

appendfsync为no，表示redis不会自动将内存中的临时数据写入磁盘中，全靠系统来进行数据保存，大多数linux系统默认为每30s保存一次缓冲池内的数据到磁盘

2.appendfsync everysec （redis默认数据写入级别）

该级别下，redis会每隔1秒进行一次数据写入磁盘操作，但如果某一次数据写入超过1秒，redis将会自动延迟一秒进行下一波写入操作，这一次写入不管花多长时间都会一直进行，但由于这时写入操作被占用，后面的写入操作将会被阻塞，导致部分数据无法正常写入；这种骚操作在大多数数据库中被称为“group commit”批量写入。

3.appednfsync always

该级别下，每一次写操作redis都会进行一次写入操作，这种级别下对数据的安全性是最好的，但由于写入次数过多，对性能会有一定影响

大佬们推介是直接使用默认级别就好