# 数据备份和恢复

## 保存数据到硬盘

1. redis 127.0.0.1:6379> SAVE
2. 127.0.0.1:6379> BGSAVE //在后台运行保存进程
3. 保存到dir目录，会生成dump.pdb数据文件

## 恢复数据

1. 将dump.pdb文件放在dir目录就好
2. Config get dir 可以获取目录路径
3. Config set dir 可以设置路径

# Redis 安全

## 查看是否设置了密码

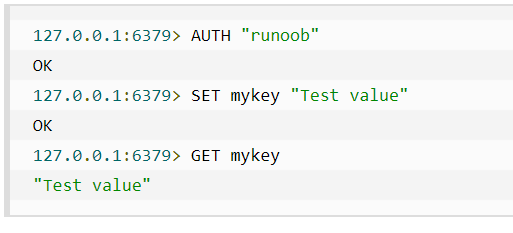
Config get requirepass

## 设置密码

Config set requirepass 123456（具体密码）

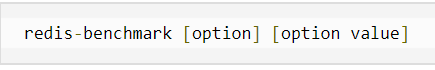
## 3, 客户端连接服务端

1. redis-cli -h IP地址 -p 端口 -a 地址
2. 在客户端中使用AUTH命令



# Redis 性能测试

## redis 性能测试的基本命令

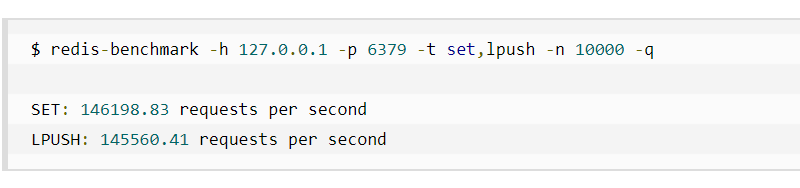


## 实例

1. 测试本机



1. 测试具体地址的redis



主机为 127.0.0.1，端口号为 6379，执行的命令为 set,lpush，请求数为 10000，通过 -q 参数让结果只显示每秒执行的请求数。

## 测试性能常用参数





# Redis 客户端连接

## 简介

Redis 通过监听一个 TCP 端口或者 Unix socket 的方式来接收来自客户端的连接，当一个连接建立后，Redis 内部会进行以下一些操作：

1. 首先，客户端 socket 会被设置为非阻塞模式，因为 **Redis 在网络事件处理上采用的是非阻塞多路复用模型。**
2. 然后为这个 socket 设置 TCP\_NODELAY 属性，禁用 Nagle 算法
3. 然后创建一个可读的文件事件用于监听这个客户端 socket 的数据发送

## 最大连接数maxclients

1. 查看 config get maxclients（默认为10000）
2. 设置 config set maxclients 10000

## 客户端命令



# Redis 管道技术

## 简介

1.1 Redis 是一种基于客户端 - 服务端模型以及请求 / 响应协议的 TCP 服务。这意味着通常情况下一个请求会遵循以下步骤：

1. 客户端向服务端发送一个查询请求，并监听 Socket 返回，通常是以阻塞模式，等待服务端响应。
2. 服务端处理命令，并将结果返回给客户端。
   1. **Redis 管道技术可以在服务端未响应时，客户端可以继续向服务端发送请求，并最终一次性读取所有服务端的响应。**
   2. 管道技术最显著的优势是提高了 redis 服务的性能。

具体应用后续再查

# Redis 分区

## 简介

1. 分区是分割数据到多个 Redis 实例的处理过程，因此每个实例只保存 key 的一个子集。

## 分区的优势

1. 通过利用多台计算机内存的和值，允许我们构造更大的数据库。
2. 通过多核和多台计算机，允许我们扩展计算能力；通过多台计算机和网络适配器，允许我们扩展网络带宽。

## 分区的不足

1. 涉及多个 key 的操作通常是不被支持的。举例来说，当两个 set 映射到不同的 redis 实例上时，你就不能对这两个 set 执行交集操作。
2. 涉及多个 key 的 redis 事务不能使用。
3. 当使用分区时，数据处理较为复杂，比如你需要处理多个 rdb/aof 文件，并且从多个实例和主机备份持久化文件。
4. 增加或删除容量也比较复杂。redis 集群大多数支持在运行时增加、删除节点的透明数据平衡的能力，但是类似于客户端分区、代理等其他系统则不支持这项特性。然而，一种叫做 presharding 的技术对此是有帮助的。

## 分区类型

Redis 有两种类型分区。 假设有 4 个 Redis 实例 R0，R1，R2，R3，和类似 user:1，user:2 这样的表示用户的多个 key，对既定的 key 有多种不同方式来选择这个 key 存放在哪个实例中。也就是说，有不同的系统来映射某个 key 到某个 Redis 服务。

* 1. 范围分区

1. 最简单的分区方式是按范围分区，就是映射一定范围的对象到特定的 Redis 实例。
2. 比如，ID 从 0 到 10000 的用户会保存到实例 R0，ID 从 10001 到 20000 的用户会保存到 R1，以此类推。
3. 这种方式是可行的，并且在实际中使用，不足就是要有一个区间范围到实例的映射表。这个表要被管理，同时还需要各 种对象的映射表，通常对 Redis 来说并非是好的方法。
   1. 哈希分区-对任何key都适用
4. 用一个 hash 函数将 key 转换为一个数字，比如使用 crc32 hash 函数。对 key foobar 执行 crc32(foobar) 会输出类似 93024922 的整数。
5. 对这个整数取模，将其转化为 0-3 之间的数字，就可以将这个整数映射到 4 个 Redis 实例中的一个了。93024922 % 4 = 2，就是说 key foobar 应该被存到 R2 实例中。注意：取模操作是取除的余数，通常在多种编程语言中用 % 操作符实现。