

# RDB

---

## RDB启动方式：save指令

命令：

```
save
```

作用：手动执行一次保存操作，会在本地生成一个以rdb结尾的文件，保存当前的快照信息。

## RDB启动方式：save指令相关配置

- dbfilename dump.rdb
  - 说明：设置本地数据库文件名，默认值为dump.rdb
  - 经验：通常设置为**dump-端口号.rdb**
- dir
  - 说明：设置存储.rdb文件的路径
  - 经验：通常设置成存储空间较大的目录，目录名称为data
- rdbcompression yes
  - 说明：设置存储本地数据库时是否压缩数据，默认为yes，采用LZF压缩
  - 经验：通常默认为开启状态，如果设置为no，可以节省CPU运行时间，但会使存储的文件变大（巨大）
- rdbchecksum yes
  - 说明：设置是否进行RDB文件格式校验，该校验过程在写文件和读文件过程均进行
  - 经验：通常默认为开启状态，如果设置为no，可以节约读写过程约10%的时间消耗，但是存储有一定的数据损坏风险

**数据恢复：RDB在服务重新启动的时候会扫描配置文件dir对应的文件夹，如果有rdb文件就会恢复数据**

**\*\*注意：\*\*save指令的执行会阻塞当前的Redis服务器，直到当前的RDB过程完成为止，如果数据量很大可能会造成长时间的阻塞，线上环境不建议使用**

## RDB启动方式：bgsave指令

## RDB启动方式 —— bgsave指令工作原理

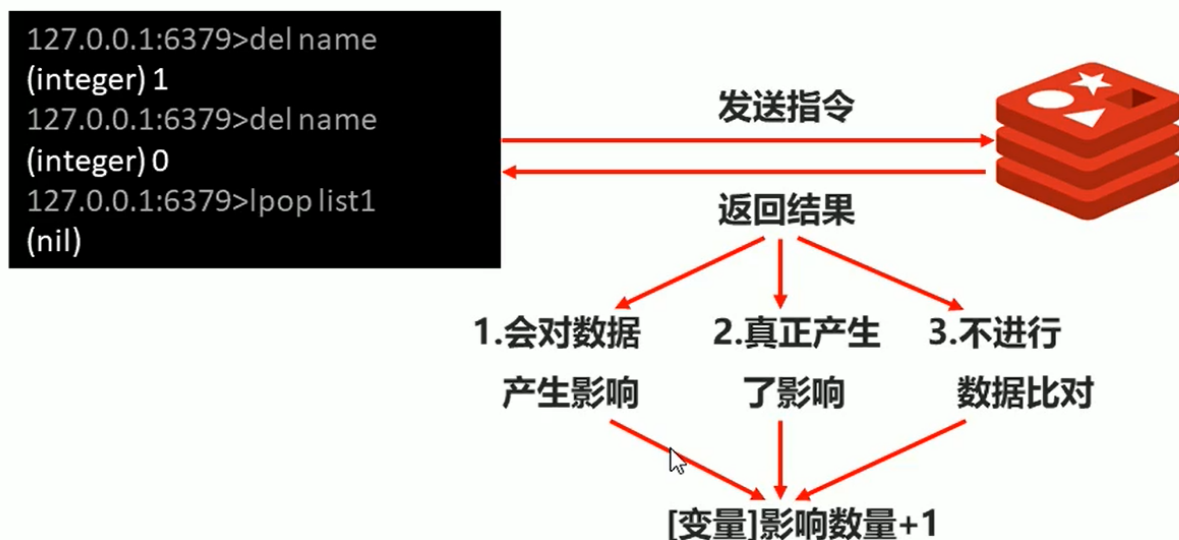


**注意:** bgsave指令是针对save阻塞问题做的优化, Redis内部所有涉及到RDB操作都采用bgsave方式。

参数配置:

- stop-writes-on-bgsave-error yes
  - 说明: 后台存储过程中如果出现错误现象, 是否停止保存操作
  - 经验: 通常默认开启状态

## RDB启动方式: save配置原理



配置:

```
save second changes
```

作用:

满足限定时间范围内可以的变换数量达到指定数量即进行持久化

参数:

- second: 监控时间范围
- changes: 监控key的变化量

#### 位置:

在conf文件中进行配置

#### 注意:

- save配置要根据实际业务情况进行设置, 频度过高或过低都会出现性能问题, 结果可能时灾难性的
- save配置中对于second与changes设置通常具有互补对应关系, 尽量不要设置成包含性关系
- save配置后台调用的时bgsave指令

## RDB两种方式对比

方式	save指令	bgsave指令
读写	同步	异步
阻塞客户端指令	是	否
额外内存消耗	否	是
启动新进程	否	是

## RDB特殊启动形式

- 全量复制 (主从复制中讲)
- 服务器运行过程中重启  
debug reload
- 关闭服务器时指定保存数据  
shutdown save

默认情况下执行**shutdown**指令时, 自动执行**bgsave** (如果没有开启AOF持久化功能)

## RDB优点

- RDB是一个紧凑压缩的二进制文件存储效率较高
- RDB内部存储的时redis在某个时间点的数据快照, 非常适合用于数据备份, 全量复制等场景
- RDB恢复数据的速度要比AOF快很多

- 应用：服务器中每X小时执行一次bgsave备份，并将rdb文件拷贝到远程机器中，用于灾难恢复

## RDB缺点

- RDB方式无论是执行指令还是利用配置，无法做到实时持久化，具有较大的可能性丢失数据
- bgsave指令每次运行要执行fork操作创建子进程，需要消耗一些性能
- Redis的众多版本中RDB文件格式不统一，有可能出现个版本服务之间数据格式无法兼容的现象