**nosql概述**

数据库类型

RDBMS

关系型数据库管理系统，按照预先设置的组织结构，将数据存储在物理介质上

数据之间可以做关联操作

RDBMS服务软件

主流的RDBMS软件

oracle

db2

ms sql server

mysql、mariadb

NOSQL

nosql(nosql=not only sql)

意思是“不仅仅是sql”

泛指非关系型数据库

不需要预先定义数据存储结构

表的每条记录都可以有不同的类型和结构

nosql服务软件

主流软件

redis

mongodb

memcached

couchdb

neo4j

flockdb

**部署redis服务**

部署redis服务

redis介绍

redis（远程字典服务器）

使用C语言编写

是一款高性能的(Key/Values)分布式内存数据库

支持数据持久化，可以把内存里的数据保存到硬盘中

也支持list、hash、set、zset数据类型

支持master-salve模式备份数据

中文网站www.redis.cn

装包

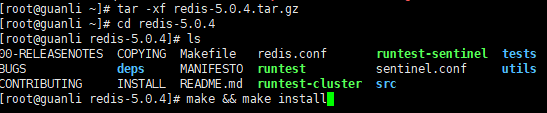
从源码包编译安装

tar -xf redis-4.0.8.tar.gz

cd redis-4.0.8

make

make install



初始化配置（初始化完成默认开启服务）

配置服务运行参数

端口：6379

主配置文件：/etc/redis/6379.conf

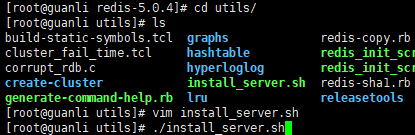
日志：/var/log/redis\_6379.log

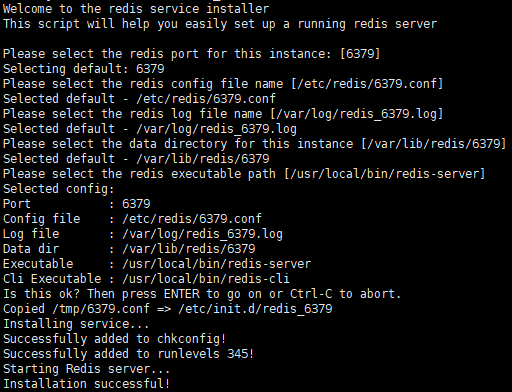
数据库目录（内存往硬盘存储的目录）：/var/lib/redis/6379

pid文件：/var/run/redis\_6379.pid

启动程序：/usr/local/bin/redis-server

./utils/install\_server.sh //初始化

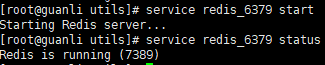




启动/停止服务

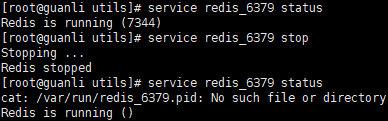
启动服务

/etc/init.d/redis\_<portnumber> start



停止服务

/etc/init.d/redis\_<portnumber> stop



连接redis数据库服务

访问redis服务

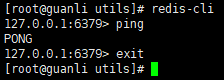
ps -C redis



netstat -utnlp | grep redis



redis-cli //连接本机的redis数据库服务



常用操作指令

set keyname keyvalue //存储

get keyname //获取

select 数据库编号0-15 //切换库

keys \* //打印所有变量

keys a? //打印指定变量

EXISTS keyname //测试是否存在

ttl keyname //查看生存时间

type keyname //查看类型

moved keyname dbname //移动变量

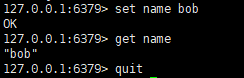
expire keyname 10 //设置有效时间

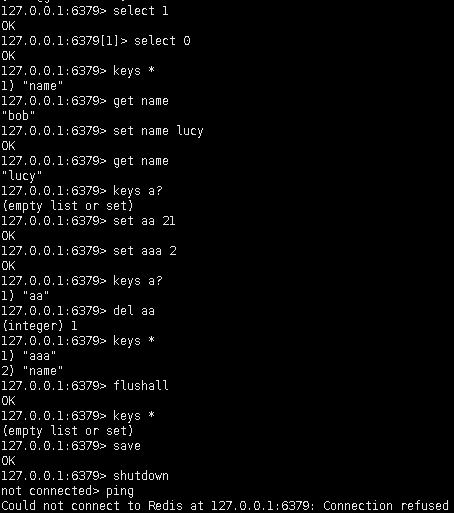
del keyname //删除变量

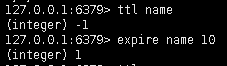
flushall //删除所有变量

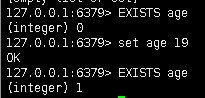
save //保存所有变量

shutdown //关闭redis服务

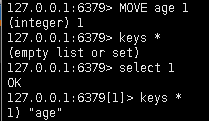








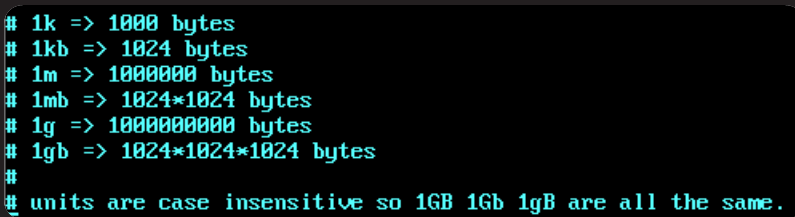




配置文件解析

数据单位

数据单位



常用配置选项

port 6379 //端口

bind 127.0.0.1 //ip地址

tcp-backlog 511 //tcp连接总数

timeout 0 //连接超时时间

tcp-keepalive 300 //长时间连接

daemonize yes //守护进程方式运行

databases 16 //数据库个数

llgfile /var/log/redis\_6379.log //pid文件

maxclients 10000 //并发连接数量

dir /var/lib/redis/6379 //数据库目录

内存管理

内存清除策略

volatile-lru //最近最少使用（针对设置了TTL的key）

allkeys-lru //删除最少使用的key

volatile-random //在设置了TTL的key里随机移除

allkeys-random //随机移除key

volatile-ttl(minor TTL) //移除最近国企的key

noeviction //不删除，写满时报错

选项默认设置

maxmemory <bytes> //最大内存

maxmemory-policy noeviction //定义使用策略

maxmemory-samples 5 //选取模版数据的个数（针对lru和ttl策略）

设置连接密码

