**MongoDB安装步骤**

1. 首先从官网上下载<https://www.mongodb.com/>上下载安装包。
2. 在安装目录下解压：tar -zxvf mongodb-linux-x86\_64-rhel62-3.4.10.tgz
3. 修改解压后名字为mongodb：mv mongodb-linux-x86\_64-rhel62-3.4.10 mongodb
4. cd mongodb目录下创建数据库文件夹和日志：

mkdir data; ->mkdir db

mkdir logs;-> touch mongodb.log

MongoDB 的可执行文件位于 bin 目录下，所以可以将其添加到 **PATH** 路径中：在/etc/profile

export PATH=<mongodb-install-directory>/bin:$PATH

<mongodb-install-directory>：是安装目录.

如：export PATH=/data00/MongoDB/mongodb/bin:$PATH

添加完成后执行source /etc/profile保存.

需要自己创建mongod.conf文件，且配置。这里我配置如下：

**[root@Amen-Sc3 bin]# more mongod.conf**

**dbpath=/data00/MongoDB/mongodb/data/db**

**logpath=/data00/MongoDB/mongodb/logs/mongodb.log**

**bind\_ip=0.0.0.0**

**port=27017**

**fork=true**

**nohttpinterface=true**

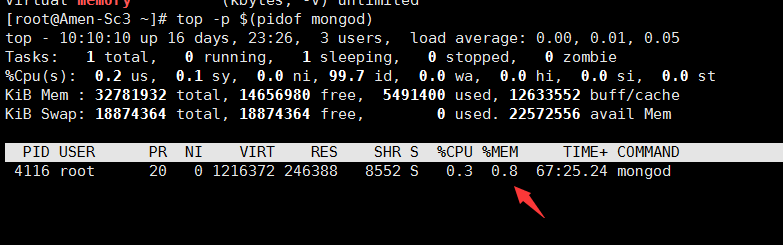
**启动命令为mongo –f mongod.conf**

**如果修改默认端口号，连接需要mongo --port=27000，不能直接使用mongo命令。**

不要使用kill命令直接杀进程，会造成数据损失。  
            建议不要使用 ”kill -9 pid“，因为如果 MongoDB 运行在没开启日志（--journal）的情况下，  
            可能会造成数据损失

在db.admin下使用db.shutdownServer()关闭服务ctrl+c退出。或者输入exit 退出

**查看mongodb的内存使用情况。命令：**top -p $(pidof mongod)；可以看到图中mongodb所占内存为0.8%。

****

如果内存过大，需要限制下mongodb的设置。

cacheSizeGB限制大小

1. **.mongod –h**:查看mongodb帮助命令。参数说明：

--logpath 日志文件路径

--master 指定为主机器

--slave 指定为从机器

--source 指定主机器的IP地址

--pologSize 指定日志文件大小不超过64M.因为resync是非常操作量大且耗时，最好通过设置一个足够大的oplogSize来避免resync(默认的 oplog大小是空闲磁盘大小的5%)。

--logappend 日志文件末尾添加

--port 启用端口号

--fork 在后台运行

--only 指定只复制哪一个数据库

--slavedelay 指从复制检测的时间间隔

--auth 是否需要验证权限登录(用户名和密码)

-h [ --help ] show this usage information

--version show version information

-f [ --config ] arg configuration file specifying additional options

--port arg specify port number

--bind\_ip arg local ip address to bind listener - all local ips

bound by default

-v [ --verbose ] be more verbose (include multiple times for more

verbosity e.g. -vvvvv)

--dbpath arg (=/data/db/) directory for datafiles 指定数据存放目录

--quiet quieter output 静默模式

--logpath arg file to send all output to instead of stdout 指定日志存放目录

--logappend appnd to logpath instead of over-writing 指定日志是以追加还是以覆盖的方式写入日志文件

--fork fork server process 以创建子进程的方式运行

--cpu periodically show cpu and iowait utilization 周期性的显示cpu和io的使用情况

--noauth run without security 无认证模式运行

--auth run with security 认证模式运行

--objcheck inspect client data for validity on receipt 检查客户端输入数据的有效性检查

--quota enable db quota management 开始数据库配额的管理

--quotaFiles arg number of files allower per db, requires --quota 规定每个数据库允许的文件数

--appsrvpath arg root directory for the babble app server

--nocursors diagnostic/debugging option 调试诊断选项

--nohints ignore query hints 忽略查询命中率

--nohttpinterface disable http interface 关闭http接口，默认是28017

--noscripting disable scripting engine 关闭脚本引擎

--noprealloc disable data file preallocation 关闭数据库文件大小预分配

--smallfiles use a smaller default file size 使用较小的默认文件大小

--nssize arg (=16) .ns file size (in MB) for new databases 新数据库ns文件的默认大小

--diaglog arg 0=off 1=W 2=R 3=both 7=W+some reads 提供的方式，是只读，只写，还是读写都行，还是主要写+部分的读模式

--sysinfo print some diagnostic system information 打印系统诊断信息

--upgrade upgrade db if needed 如果需要就更新数据库

--repair run repair on all dbs 修复所有的数据库

--notablescan do not allow table scans 不运行表扫描

--syncdelay arg (=60) seconds between disk syncs (0 for never) 系统同步刷新磁盘的时间，默认是60s

Replication options:

--master master mode 主复制模式

--slave slave mode 从复制模式

--source arg when slave: specify master as <server:port> 当为从时，指定主的地址和端口

--only arg when slave: specify a single database to replicate 当为从时，指定需要从主复制的单一库

--pairwith arg address of server to pair with

--arbiter arg address of arbiter server 仲裁服务器，在主主中和pair中用到

--autoresync automatically resync if slave data is stale 自动同步从的数据

--oplogSize arg size limit (in MB) for op log 指定操作日志的大小

--opIdMem arg size limit (in bytes) for in memory storage of op ids指定存储操作日志的内存大小

Sharding options:

--configsvr declare this is a config db of a cluster 指定shard中的配置服务器

--shardsvr declare this is a shard db of a cluster 指定shard服务器

@Id - 文档的唯一标识，在mongodb中为ObjectId，它是唯一的，通过时间戳+机器标识+进程ID+自增计数器（确保同一秒内产生的Id不会冲突）构成。

@Document - 把一个java类声明为mongodb的文档，可以通过collection参数指定这个类对应的文档。@Document(collection="mongodb") mongodb对应表

@DBRef - 声明类似于关系数据库的关联关系。ps：暂不支持级联的保存功能，当你在本实例中修改了DERef对象里面的值时，单独保存本实例并不能保存DERef引用的对象，它要另外保存，如下面例子的Person和Account。

@Indexed - 声明该字段需要索引，建索引可以大大的提高查询效率。

@CompoundIndex - 复合索引的声明，建复合索引可以有效地提高多字段的查询效率。

@GeoSpatialIndexed - 声明该字段为地理信息的索引。

@Transient - 映射忽略的字段，该字段不会保存到mongodb。

@PersistenceConstructor - 声明构造函数，作用是把从数据库取出的数据实例化为对象。该构造函数传入的值为从DBObject中取出的数据

./mongod -f mongod.conf

./mongo --port 27000

执行exit退出。

**./mongodump -u boncamen -p boncPW.123 -h 127.0.0.1:27000 -d spacetime2 -o /home/mongodump** //备份整个mongodb

备份单个集合 （-o 后面需要跟目录和自定义集合名）

**./mongoexport -u boncamen -p boncPW.123 -h 127.0.0.1:27000 -d spacetime2 -c basestation -o /home/mongodump/basestation**

**./mongorestore -h 127.0.0.1:27000 -d spacetime2 /data01/spacetime2 // 恢复整个mongodb数据**

导入单个集合

**./mongoimport -u boncamen -p boncPW.123 -h 127.0.0.1:27000 -d spacetime2 -c basestation --file /data01/local/basestation**

**./mongostat --host 127.0.0.1:27000 -uroot -p1qaz2wsx --authenticationDatabase admin (监控mongodb运行状态)**

db.createUser(

{

user: "root",

pwd: "1qaz2wsx",

roles: [ { role: "root", db: "admin" } ]

}

)

db.createUser(

{

user: "boncamen",

pwd: "boncPW.123",

roles: [ { role: "readWrite", db: "spacetime2" } ]

}

)

