# Mongodb安装

## Mongodb环境

安装环境

安装环境是64位centos 6.5。使用二进制文件mongodb-linux-x86\_64-3.0.6.tgz。

安装所需材料在“mongodb部署材料”下。

## 安装并配置

### 解压和重命名

* 解压

# tar zxvf mongodb-linux-x86\_64-3.0.6.tgz

* 移动并重命名（我的安装路径是：/usr/local）

# mv mongodb-linux-x86\_64-3.0.6 /usr/local/mongodb3

### 启动文件配置

* 环境变量（可以不配置）

**注意：需要手动创建data和logs连个文件夹，不然启动会失败**

export MONGODB\_HOME=/usr/local/mongodb3/bin

export PATH=$MONGODB\_HOME:$PATH

* 创建数据和日志文件夹

**注意：这里我创建到/目录下，如果没有权限，启动会报error number 1，则创建到有权限的路径中，也可以正常的也可以创建到**/usr/local/mongodb/data/db，/usr/local/mongodb/data/logs/mongoddb.log，**配置文件也可以创建到**/usr/local/mongodb3/conf/mongodb.conf

# mkdir –p /data/db

# mkdir –p /data/logs

# touch mongodb.log

* 创建启动配置文件

# vim /usr/local/mongodb3/bin/mongodb.conf

配置内容如下，其他详细配置参看附录

dbpath=/data/db #必须手动创建该路径

logpath=/data/logs/mongodb.log #必须手动创建该文件（必须是文件）

pidfilepath = /var/run/mongo.pid #pid文件，不用手动创建

logappend=true #日志输出 true 追加 false 每次启动另创日志文件

bind\_ip=0.0.0.0 #绑定IP 默认127.0.0.1，这个ip放行用，非

port=27017 #注册本地端口

fork=true #后台启动

如果采用互为副本配置需要增加配置

replSet=ngqualitymdl/127.0.0.1:27017 #MongoDB的副本集

然后使用命令mongod访问任意一台（最好是引擎使用的那台）

> config = {\_id: 'ngqualitymdl', members: [

{"\_id":1,"host":"127.0.0.1:27017"},

{"\_id":2,"host":"127.0.0.1:27027"}

]}

> rs.initiate(config);

使用rs.status();查看状态即可。

### 防火墙放行(注意)

* Centos 7.0

# firewall-cmd --zone=public --add-port=27017/tcp –permanent

# firewall-cmd restart

* Centos 6.0

# vim /etc/sysconfig/iptables

放行端口，配置到22端口前面

-A INPUT -m state --state NEW -m tcp -p tcp --dport 27017 -j ACCEPT

重启防火墙

# service iptables restart

如果没有放行27017端口，下面搭建mongodb副本集会出现问题

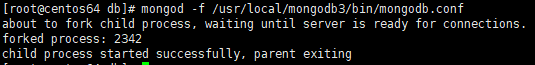
## 简单测试

* 在任意位置查看mongodb版本,查询到结果，则环境变量配置成功

# mongod –version

* 然后测试启动

# mongod –f /usr/local/mongodb3/bin/mongodb.conf

如图所示启动成功

* 关闭软件

# mongod –f /usr/local/mongodb3/bin/mongodb.conf –shutdown

错误信息：



日志文件指向的是文件夹，mongdb.log是文件不是文件夹，

关于error number 1 大多数是权限和路径的问题，

Error number 100 是mongodb没有正常关闭

注意：不要用kill -9 杀死进程

## 开机启动服务(2种方式)

### 添加到/etc/init.d/（推荐）

* 创建mongodb服务脚本（已提供脚本mongod，可以直接添加到路径/etc/init.d/下）

# vim /etc/init.d/mongod

* 服务配置自启动：

逐条执行下面命令：

# chkconfig --add mongod

# chmod u+x  mongod

# chkconfig mongod on

* 粘贴如下代码到mongodb服务脚本

#!/bin/sh

# chkconfig: 2345 93 18

#MogoDB home directory

MONGODB\_HOME=/usr/local/mongodb3

#mongodb command

MONGODB\_BIN=$MONGODB\_HOME/bin/mongod

#mongodb config file

MONGODB\_CONF=$MONGODB\_HOME/bin/mongodb.conf

#mongodb PID

MONGODB\_PID=/var/run/mongo.pid

#set open file limit

SYSTEM\_MAXFD=65535

MONGODB\_NAME="mongodb"

. /etc/rc.d/init.d/functions

if [ ! -f $MONGODB\_BIN ]

then

echo "$MONGODB\_NAME startup: $MONGODB\_BIN not exists! "

exit

fi

start(){

ulimit -HSn $SYSTEM\_MAXFD

$MONGODB\_BIN --config="$MONGODB\_CONF" --fork ##added @20160901

ret=$?

if [ $ret -eq 0 ]; then

action $"Starting $MONGODB\_NAME: " /bin/true

else

action $"Starting $MONGODB\_NAME: " /bin/false

fi

}

stop(){

PID=$(ps aux |grep "$MONGODB\_NAME" |grep "$MONGODB\_CONF" |grep -v grep |wc -l)

if [[ $PID -eq 0 ]];then

action $"Stopping $MONGODB\_NAME: " /bin/false

exit

fi

kill -HUP `cat $MONGODB\_PID`

ret=$?

if [ $ret -eq 0 ]; then

action $"Stopping $MONGODB\_NAME: " /bin/true

rm -f $MONGODB\_PID

else

action $"Stopping $MONGODB\_NAME: " /bin/false

fi

}

restart() {

stop

sleep 2

start

}

case "$1" in

start)

start

;;

stop)

stop

;;

status)

status $prog

;;

restart)

restart

;;

\*)

echo $"Usage: $0 {start|stop|status|restart}"

esac

### 添加到/etc/rc.local

**注意：路径不要用相对路径，推荐使用步骤1.3.2添加为mongod服务**

* 编辑 /etc/rc.local

# vim /etc/rc.local

* 文件添加如下内容

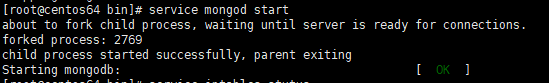
/usr/local/mongodb3/bin/mongod -f /usr/local/mongodb3/bin/mongodb.conf

## 测试服务

启动

# service mongod start

如图是成功状态



重启

# service mongod restart

停止

#service mongod stop

# 启动、关闭以及故障修复

## 关闭（重要）

**注意： mongodb不要用kill -9 pid 去杀死进程,可以使用kill -2 pid，否则可能导致数据丢失和数据文件损坏（需要修复），而且下次启动会失败**。

方式一：

# mongod –f /usr/local/mongodb3 –shutdown

方式二：

# ps aux | grep mongod

# kill -2 pid

## 启动

方式一：

# mongod –f /usr/local/mongod3/bin/mongodb.conf

方式二（推荐）：

需要配置mongodb服务

# service mongod start

## 故障修复

**注意：若数据库出现如不能连上，则是一个data目录下的mongod.lock文件的问题，可以用如下的修复的命令，**

# mongod –repair

# mongod –f /usr/local/mongod3/bin/mongodb.conf

# 数据备份

## 导入8个规则

**注意：使用我们提供的jar包导入更新8个规则，执行GrammarParser文件夹下jar包，命令如下：**

# java –jar ./ GrammarParser-1.0-SNAPSHOT.jar

## 导出数据

进入mongodb安装路径 mongodb/bin，执行下面命令

* 语法

# mongoexport -d dbname -c collectionname -o file --type json/csv -f field

* 示例

# sudomongoexport –d mongotest-cusers -o /home/mongoDB/users.json --type json -f  "\_id,user\_id,user\_name,age,status"

参数说明：

-d ：数据库名

-c ：collection名

-o ：输出的文件名

--type ： 输出的格式，默认为json

-f ：输出的字段，如果-type为csv，则需要加上-f "字段名"

## 导入数据

* 语法

# mongoimport -d dbname -c collectionname --file filename --headerline --type json/csv -f field

* 示例

# sudo mongoimport -d mongotest -c users --file /home/mongodump/articles.json --type json

参数说明：

     -d ：数据库名

     -c ：collection名

  --type ：导入的格式默认json

     -f ：导入的字段名

    --headerline ：如果导入的格式是csv，则可以使用第一行的标题作为导入的字段

    --file ：要导入的文件

## 备份

* 语法

# mongodump -h dbhost -d dbname -o dbdirectory

* 实例

# sudo rm -rf /home/momgodb/backup

# sudo mkdir -p /home/momgodb/backup

# sudo mongodump -h 192.168.109.134:27017 -d test -o /home/mongodb/backup /

 参数说明：

-h： MongDB所在服务器地址，例如：127.0.0.1，当然也可以指定端口号：127.0.0.1:27017

-d： 需要备份的数据库实例，例如：test

-o： 备份的数据存放位置，例如：/home/mongodump/，当然该目录需要提前建立，这个目录里面存放该数据库实例的备份数据。

## 恢复

* 语法：

# mongorestore -h dbhost -d dbname --dir dbdirectory

* 实例：

# mongorestore -h 192.168.109.134:27017 -d test2 –dir /home/mongodb/backup/

参数或名：

-h： MongoDB所在服务器地址

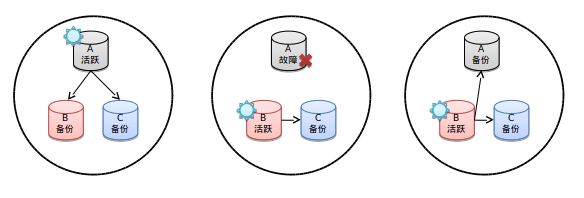
-d： 需要恢复的数据库实例，例如：test，当然这个名称也可以和备份时候的不一样，比如test2

--dir： 备份数据所在位置，例如：/home/mongodb/backup

--drop： 恢复的时候，先删除当前数据，然后恢复备份的数据。就是说，恢复后，备份后添加修改的数据都会被删除，慎用！

# 副本集搭建（重要）

原理图;



副本集有点类似主从复制，不过跟真正的主从复制还是有两点区别的。

* 该集群没有特定的主数据库。
* 如果哪个主数据库宕机了，集群中就会推选出一个从属数据库作为主数据库顶上，这就具备了自动故障恢复功能.
* 该集群没有特定的主数据库。
* 如果哪个主数据库宕机了，集群中就会推选出一个从属数据库作为主数据库顶上，这就具备了自动故障恢复功能.

目前副本集有由来两台服务器组成

## 配置mongodb.conf

机器一(192.168.109.137)：

在mongodb.conf追加如下配置

replSet = ngqualitymdl #设定同伴 child为集群名称

机器二(192.168.109.138)：

replSet = ngqualitymdl

机器二(192.168.109.139)：

replSet = ngqualitymdl

## 清理数据库（重要）

搭建副本集需要确保SECONDARY（不活跃状态）数据结构是PRIMARY（活跃状态）的子集，如果不确定建议清空所有mongodb副本数据。

## 按顺序启动副本集

* 启动机器一
* 启动机器二
* 选中一台机器开始初始化副本集，初始化命令

提醒;这里可以配置priority，一般建议搭建副本集机器数是奇数台

> config\_rs1={\_id:'ngqualitymdl',members:[

{\_id:0,host:'192.168.109.137:27017',priority:10},

{\_id:1,host:'192.168.109.138:27017',priority:9},

{\_id:1,host:'192.168.109.139:27017',priority:8}]

}

> rs.initiate(config\_rs1)

如图是正确初始化结果



* 使用rs.status();查看状态即可。

## 测试验证（重要）

**主从服务器数据是否同步，从服务器没有读写权限，服务器是大于3的奇数**

* a:向主服务器写入数据 ok 后台自动同步到从服务器，从服务器有数据
* b:向从服务器写入数据 false 从服务器不能写
* c:主服务器读取数据 ok
* d:从服务器读取数据 false 从服务器不能读

**关闭主服务器，从服务器是否能顶替**   
此时你关掉活跃节点的服务。此时你会发现剩余的两台机器有一台变为活跃节点了。

## 配置副本集的其他配置参数

**节点和初始化高级参数**

* standard 常规节点:参与投票有可能成为活跃节点
* passive 副本节点:参与投票,但是不能成为活跃节点
* arbiter 仲裁节点:只是参与投票不复制节点也不能成为活跃节点

**高级参数**

* Priority 0到1000之间 ,0代表是副本节点 ,1到1000是常规节点
* arbiterOnly : true 仲裁节点

# 附录

## 配置文件参数详解

--dbpath 数据库路径(数据文件)

--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)

## 启动参数详解

-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，默认路径/data/db/

--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