# MyCat安装

去官网下载安装包：

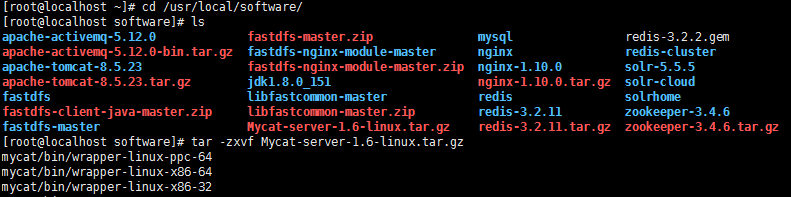


下载路径：

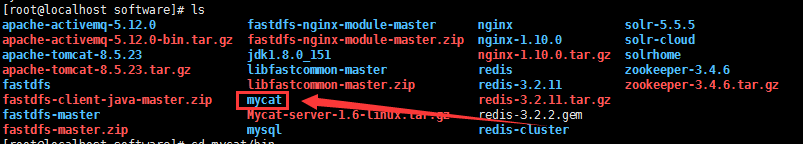
官方网站：<http://www.mycat.org.cn>/

github地址<https://github.com/MyCATApache>

上传到服务器上并解压：



它解压后是一个叫mycat的文件夹

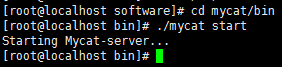


去maycat的bin目录下，执行命令

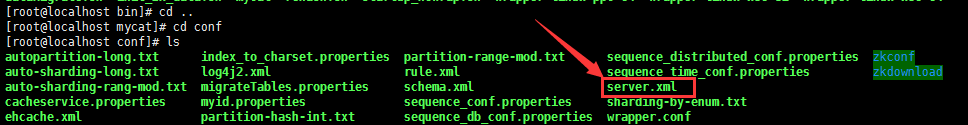
./mycat start 启动服务

./mycat stop 停止服务

./mycat restart 重启服务



mycat的conf文件夹下有一个server.xml，里面有，mycat连接的用户名和密码

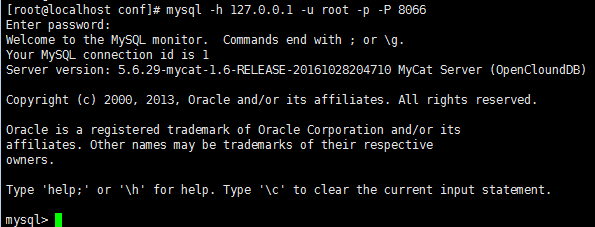




可以看到root用户密码是123456

连接使用mysql客户端

mysql -h 127.0.0.1 -u root -p -P 8066

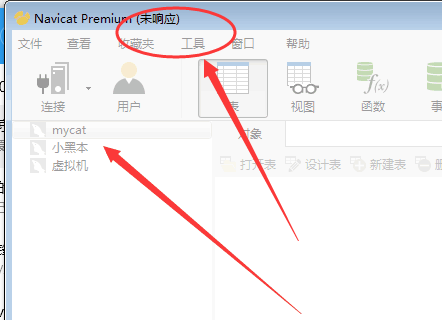


远程连接，记得防火墙开放8066端口

使用iptables开放如下端口  
/sbin/iptables -I INPUT -p tcp --dport 8066 -j ACCEPT  
保存  
/etc/rc.d/init.d/iptables save  
重启服务  
service iptables restart

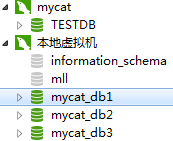


注意新安装的mycat，再使用远程工具的时候可能会卡住，多等一会就好了。



# 分片

在mysql连接里建立三个数据库，注意是从前我们用的mysql连接，不是mycat。

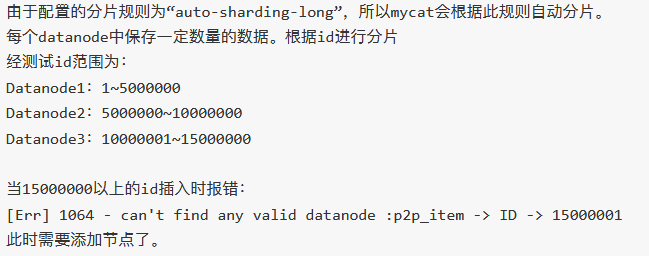


修改mycat/conf/schema.xml

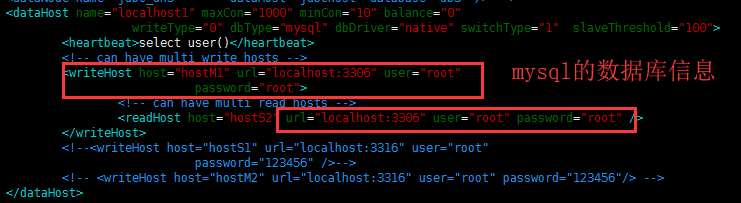


<table>用来配置表，原来的配置文件中有许多测试用的表。我们追加两个表，test\_tb01和test\_tb02。

test\_tb01放到三个节点上，分别对应新创建的三个库mycat\_db1、mycat\_db2、 mycat\_db3。rule指定它的分片策略是按照id分

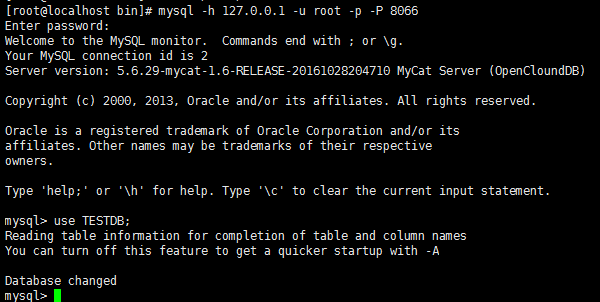


分片规则：<https://blog.csdn.net/u011731544/article/details/80579088>



test\_tb02的type=global，代表全局表

改完重启mycat



执行语句

DROP TABLE IF EXISTS `test\_tb01`;

CREATE TABLE `test\_tb01` (

`ID` int(11) NOT NULL AUTO\_INCREMENT,

`NAME` varchar(255) DEFAULT NULL,

PRIMARY KEY (`ID`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO `test\_tb01`(id,name) VALUES ('5000001', 'aaaa');

INSERT INTO `test\_tb01`(id,name) VALUES ('1', 'aaaa');

INSERT INTO `test\_tb01`(id,name) VALUES ('2', 'aaaa');

INSERT INTO `test\_tb01`(id,name) VALUES ('10000001', 'aaaa');

DROP TABLE IF EXISTS `test\_tb02`;

CREATE TABLE `test\_tb02` (

`ID` int(11) NOT NULL AUTO\_INCREMENT,

`NAME` varchar(255) DEFAULT NULL,

PRIMARY KEY (`ID`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO `test\_tb02` (name)VALUES ('cccc');

注意表名的大小写要和配置文件中一致。

执行之后可以看到test\_tb01的数据，根据id的不同，存储到了三个节点中。

test\_tb02的数据在两个节点中是同步的。

启动mysql

cd /usr/local/mysql/support-files  
[root@localhost support-files]# ./mysql.server start

# 读写分离

数据库读写分离对于大型系统或者访问量很高的互联网应用，是必不可少的一个重要功能。对于MySql来说，标准的读写分离是主从模式，一个写节点Master跟着多个读节点。读节点的数量取决于系统的压力。通常1-3个读节点。

MyCat的读写分离需要Mysql的主从复制机制配合。

MySql主从复制需要注意的是：

1. 主DB Server和从DB Server的数据库版本一致
2. 主DB Server和从DB Server的数据库数据一致
3. 主DB Server开启二进制日志，主DB Server和从DB Server的server\_id必须唯一

## 3.1 主库配置

修改主库的/etc/my.cnf,追加如下内容：

binlog-do-db=mycat\_db1

binlog-do-db=mycat\_db2

binlog-do-db=mycat\_db3

binlog-ignore-db=mysql

log-bin=mysql-bin

server-id=144



binlog-do-db是要同步的库

binlog-ignore-db是不同步的库

log-bin开启二进制日志

server-id必须是唯一的服务号，可以写成ip最后一段

重启mysql

service mysql restart

登录mysql，创建备份用户并授权

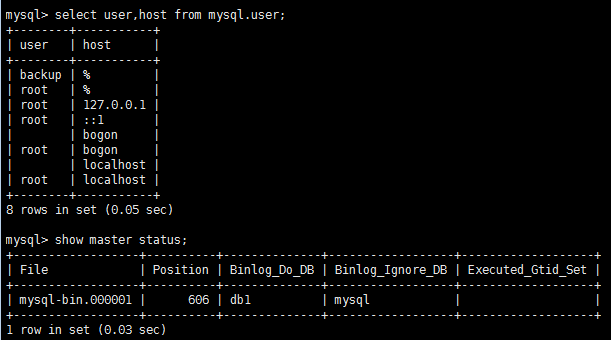
grant file on \*.\* to 'backup' @'%' identified by '123456';

grant replication slave,replication client on \*.\* to 'backup' @'%' identified by '123456';

flush privileges;

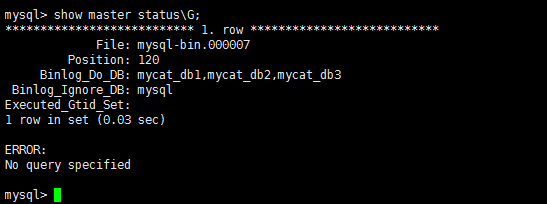
查看现在有哪些用户：

select user,host from mysql.user;



查看master状态

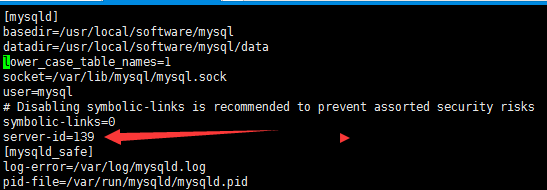
show master status\G;



## 3.2 配置从库

修改从库的/etc/my.cnf

server-id=139 可以使用ip的最后一段



重启服务

service mysql start

连接mysql,配置master的信息，注意这是一条语句，中间不要断开

CHANGE MASTER TO MASTER\_HOST='10.0.107.145', MASTER\_USER='backup',MASTER\_PASSWORD='123456',MASTER\_LOG\_FILE='mysql-bin.000007',MASTER\_LOG\_POS=120;



其中MASTER\_HOST是主库的ip,MASTER\_USER和MASTER\_PASSWORD是同步时的账号及密码，就是3.1中创建并授权的账号。MASTER\_LOG\_FILE是3.1中show master status时的file。

MASTER\_LOG\_POS 是show master status时的Position。

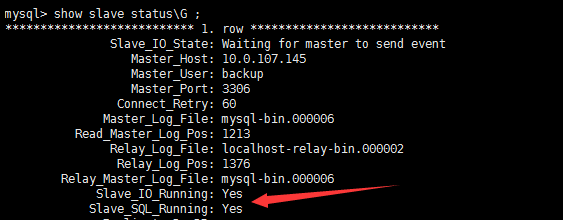
启动slave

start slave;



查看从服务器状态：

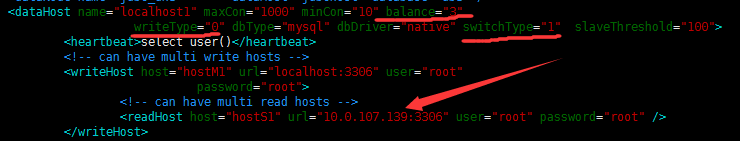
show slave status\G;



Slave\_IO\_Running: Yes //必须是YES

Slave\_SQL\_Running: Yes //必须是YES

## 3.3 配置mycat



balance:

balance="0", 不开启读写分离机制，所有读操作都发送到当前可用的writeHost 上。

balance="1"，全部的 readHost 与 stand by writeHost 参与 select 语句的负载均衡，简单的说，当双主双从模式(M1 ->S1 ， M2->S2，并且 M1 与 M2 互为主备)，正常情况下， M2,S1,S2 都参与 select 语句的负载均衡。

balance="2"，所有读操作都随机的在 writeHost、 readhost 上分发。

balance="3"， 所有读请求随机的分发到 wiriterHost 对应的 readhost 执行,writerHost 不负担读压力，注意 balance=3 只在 1.4 及其以后版本有， 1.3 没有。

writeType:

writeType="0", 所有写操作发送到配置的第一个 writeHost，第一个挂了切到还生存的第二个writeHost，重新启动后已切换后的为准，切换记录在配置文件中:dnindex.properties .

writeType="1"，所有写操作都随机的发送到配置的 writeHost。

writeType="2"，没实现。

switchType:

-1 表示不自动切换

1 默认值，自动切换

2 基于MySQL 主从同步的状态决定是否切换

重启mycat服务。

# java连接mycat

数据库链接地址，连接服务器的8066端口，指定连接到TESTDB库，用户名密码是mycat的server.xml中配置的用户名密码。其它操作和连接mysql一样：

**jdbc:mysql://www.vm.com:8066/TESTDB?characterEncoding=utf-8**