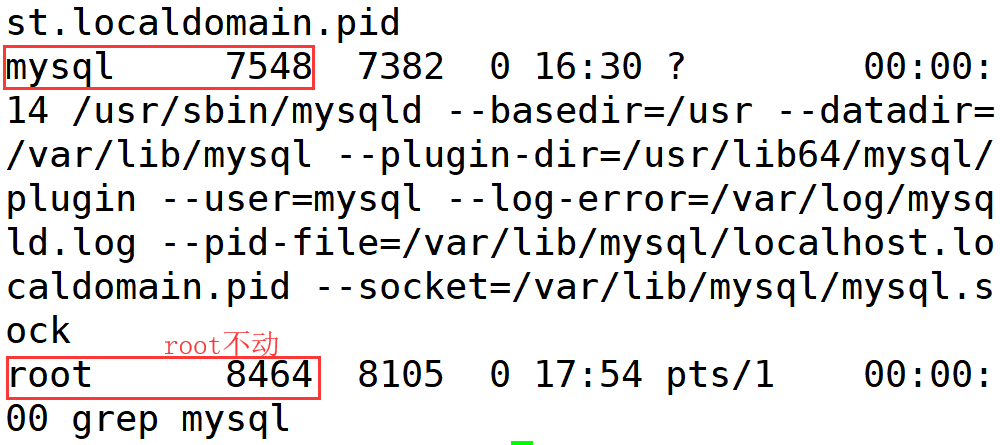
# 数据库启动问题

## 数据重启

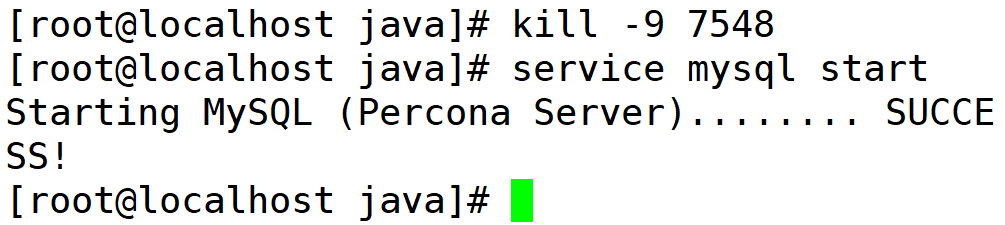
### 启动报错PID

如果mysql启动异常,则一定是操作异常,需要通过linux命令查询mysql服务的进程号.

命令:ps -ef |grep mysql



关闭进程: **kill -9 7548**



说明:mysql服务停止,则需要重启tomcat服务器.

# 实现数据库优化

## 数据库主从

### 需求说明

说明.现在如果只有一台数据库,如果这台数据库宕机,则直接影响整个服务运行.所以需要多台数据库

### 数据备份

1. 数据库冷备份

说明:定期将数据库中的到.sql文件中.之后将.sql文件中的数据导入另外的数据库中.

缺点:

1.不能实时备份数据,可能丢数据

2.备份操作复杂.

3.该操作不易实现.

记住:冷备份是恢复数据最后有效的手段

### 数据库热备份

核心思想:可以实现数据库的实时备份



实现步骤:

1. 当主库做”更新”操作时,将主库的操作全部记录到主库的二进制文件中.
2. 从库通过IO线程,读取主库的二进制日志文件,并且将更新的信息获取到之后写入从库的中继日志中.
3. 之后从库通过sql线程,将中继日志中的数据取出后,最终同步到从库中.至此主从实现了数据的同步!!

注意事项:

1. 主库中的**二进制日志文件**默认是关闭的.所以需要手动的开启
2. 注意标示符,指定同步的位置.
3. 最终从库的线程全部启动即可!!(2个yes)

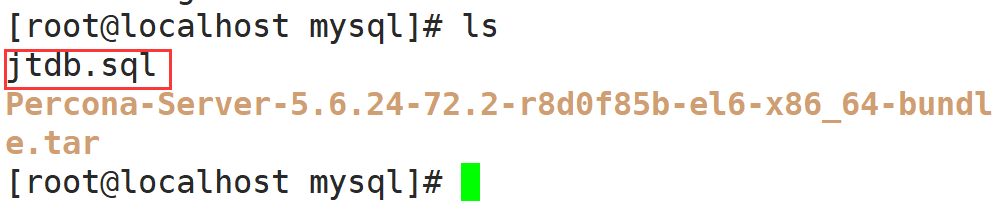
## 实现数据库主从同步

### 数据库搭建说明

1. 在原有基础上,克隆一个全新的虚拟机,安装数据库充当从库.

### 数据库安装

1. 导入jtdb.sql文件



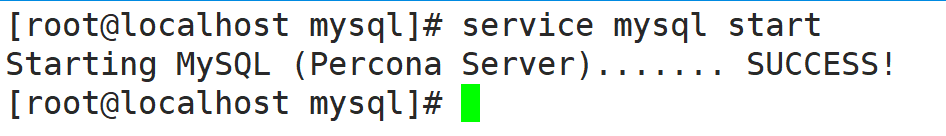
1. 解压安装包

tar -xvf Percona-Server-5.6.24-72.2-r8d0f85b-el6-x86\_64-bundle.tar

1. 安装顺序
2. 56-debuginf0
3. Shared
4. Client
5. Server

rpm -ivh Percona-Server-56-debuginfo-5.6.24-rel72.2.el6.x86\_64.rpm

4.启动数据库



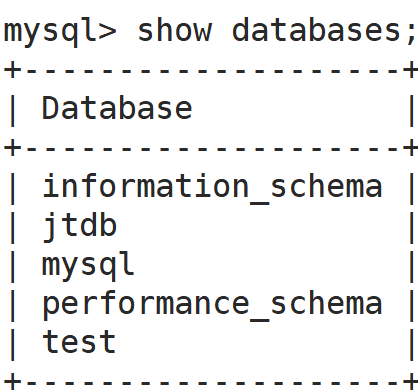
5.设定用户名和密码

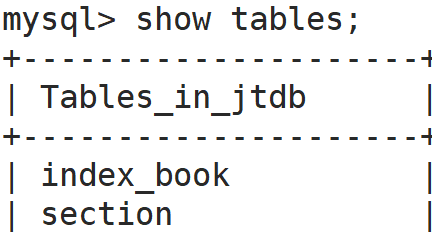
mysqladmin -u root password root

6.导入数据库文件

source jtdb.sql;

7.检查数据库是否导入成功





8.开放远程权限

mysql> grant all on \*.\* to "root"@"%" identified by "root";

Query OK, 0 rows affected (0.26 sec)

9.关闭防火墙

service iptables stop

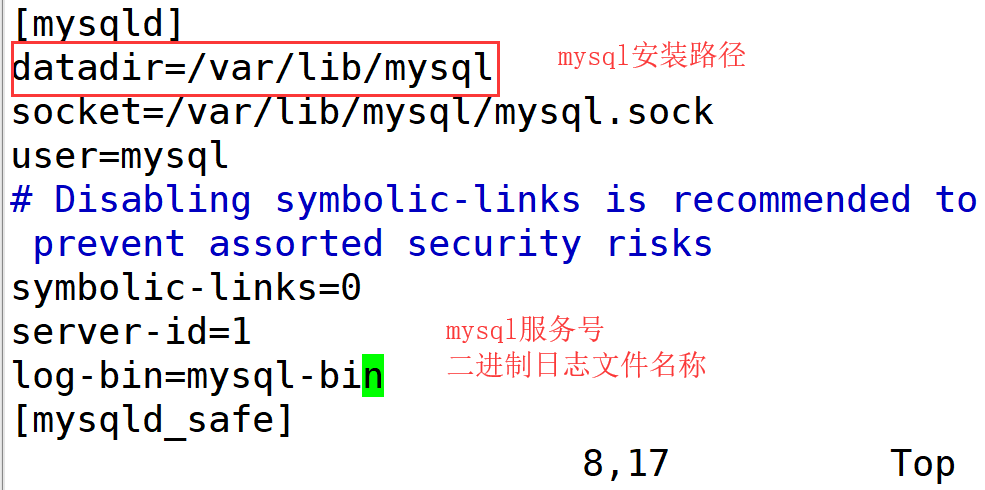
8.远程连接



## 实现主从挂载

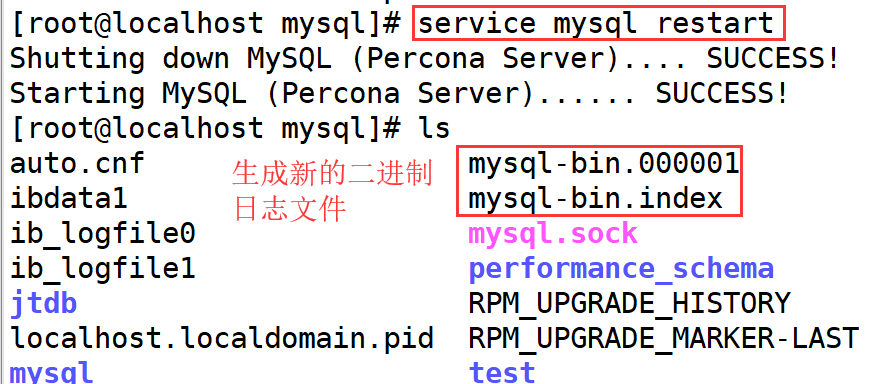
### 开启二进制日志文件

路径:vim /etc/my.cnf



### 重启mysql

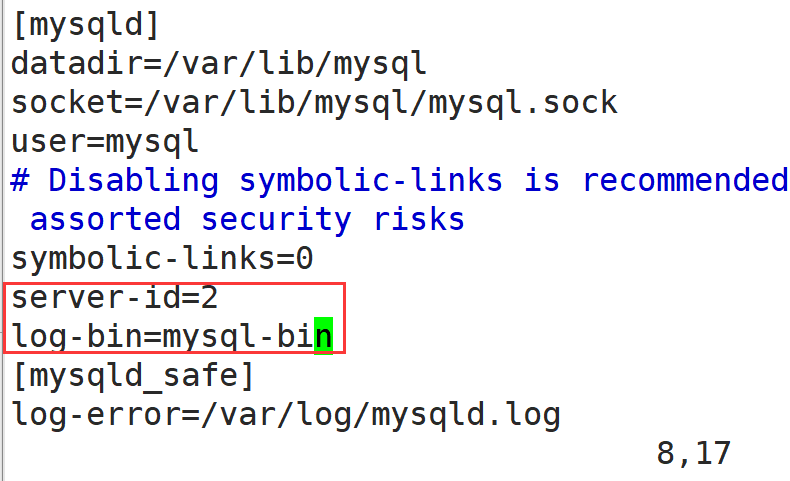
service mysql restart



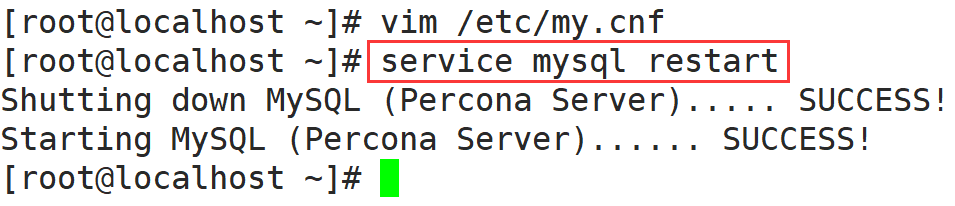
### 配置从库

1. 修改从库配置文件

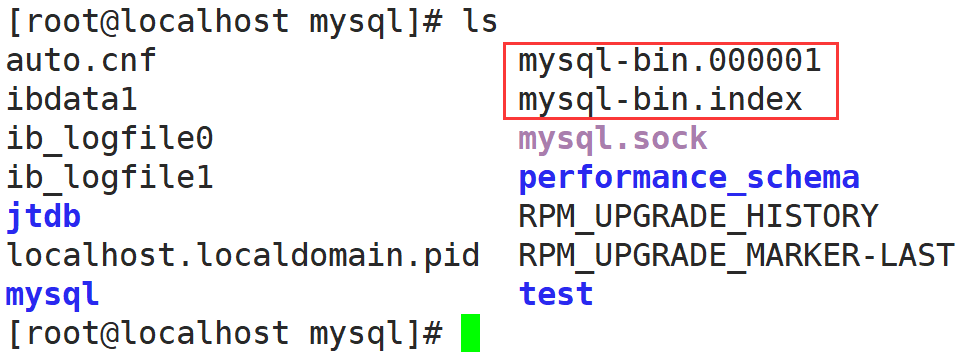
vim /etc/my.cnf



1. **重启从库**

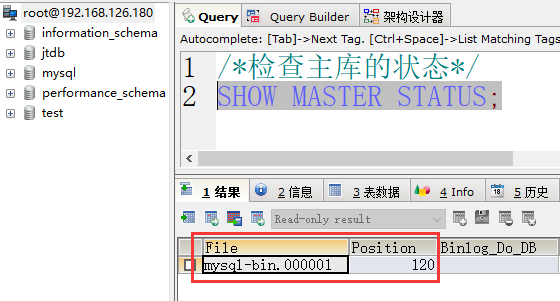


1. 检查mysql文件状态



## 实现主从同步

### 检查主库的状态



### 实现主从同步

1. 命令

/\*我是从库163 实现主从挂载 主库ip,端口,用户名,密码,二进制日志文件,位置\*/

CHANGE MASTER TO MASTER\_HOST="192.168.126.180",

MASTER\_PORT=3306,

MASTER\_USER="root",

MASTER\_PASSWORD="root",

MASTER\_LOG\_FILE="mysql-bin.000001",

MASTER\_LOG\_POS=120;

#启动主从同步的服务

START SLAVE;

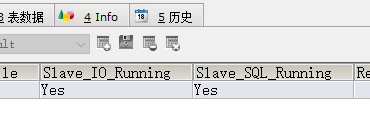
#检查主从同步的状态

SHOW SLAVE STATUS;

#如果程序执行错误 先应该关闭主从服务,之后重新挂载主机

STOP SLAVE;

### 检查主从状态



## Amoeba实现读写分离/负载均衡

### Amoeba介绍

Amoeba是一个以MySQL为底层数据存储，并对应用提供MySQL协议接口的proxy。**它集中地响应应用的请求，依据用户事先设置的规则**，将SQL请求发送到特定的数据库上执行。基于此可以实现**负载均衡、[读写分离](https://baike.baidu.com/item/%E8%AF%BB%E5%86%99%E5%88%86%E7%A6%BB" \t "_blank)、高可用性**等需求。与MySQL官方的MySQL Proxy相比，作者强调的是amoeba配置的方便（基于XML的配置文件，用SQLJEP语法书写规则，比基于lua脚本的MySQL Proxy简单）。

Amoeba相当于一个SQL请求的[路由器](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E5%99%A8" \t "_blank)，目的是为负载均衡、读写分离、高可用性提供机制，而不是完全实现它们。用户需要结合使用MySQL的 Replication等机制来实现副本同步等功能。amoeba对底层数据库连接管理和路由实现也采用了可插拨的机制，第三方可以开发更高级的策略类来替代作者的实现。这个程序总体上比较符合[KISS原则](https://baike.baidu.com/item/KISS%E5%8E%9F%E5%88%99" \t "_blank)的思想。

### Amoeba安装配置

0.amoeba的运行需要依赖JDK

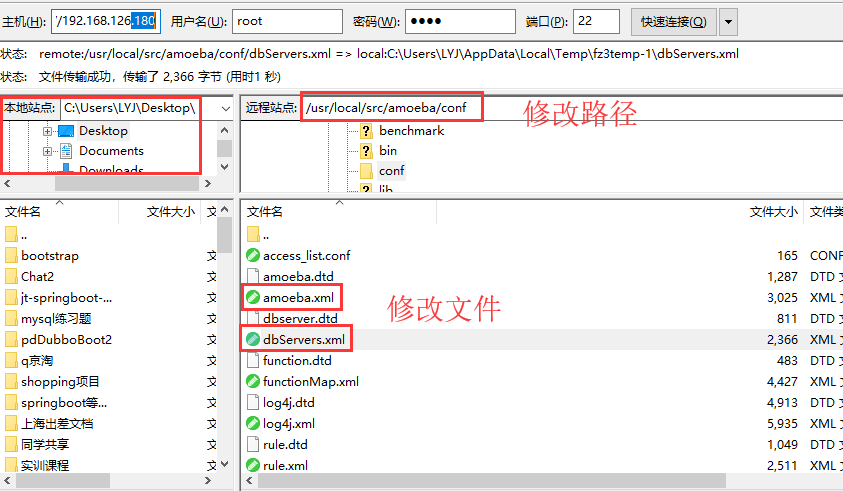
1. 上传amoeba的安装文件
2. 解压amoeba

tar -xvf amoeba-mysql-3.0.4-BETA.tar.gz

1. 修改amoeba文件名称

mv amoeba-mysql-3.0.4-BETA amoeba

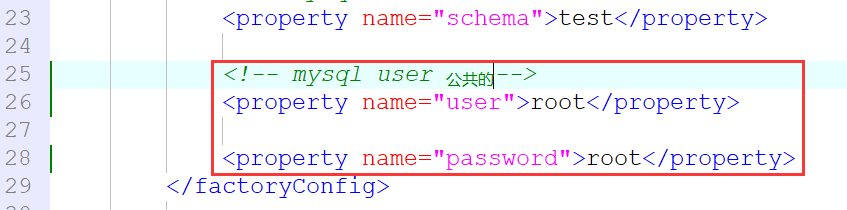
### 通过工具远程操作文件



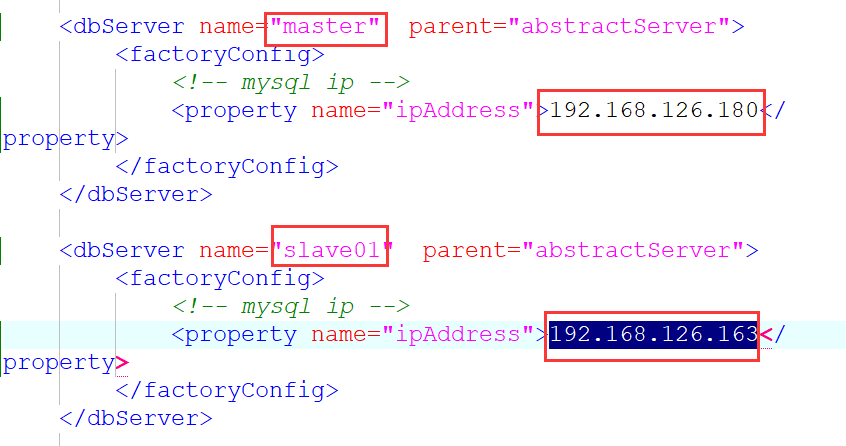
### 编辑dbServers.xml

说明:该配置文件中配置的是amoeba连接数据的策略,IP/端口/用户名/密码.

1. 编辑用户名和密码



1. 编辑数据库链接



1. 定义读的池

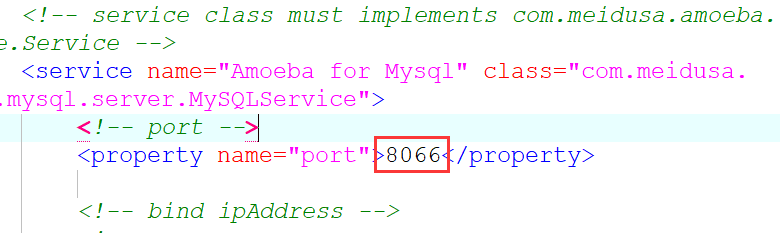


注意:”,”号的写法!!!!.

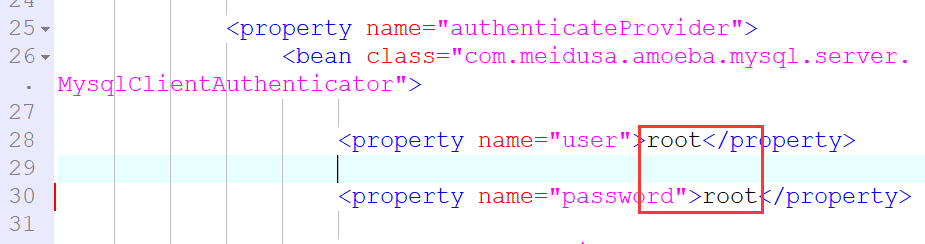
修改完成后,将文件上传到amoeba的conf文件下.

### 编辑Amoeba.xml

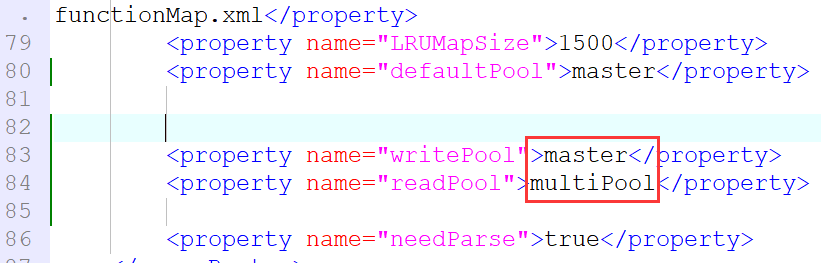
1. 定义端口



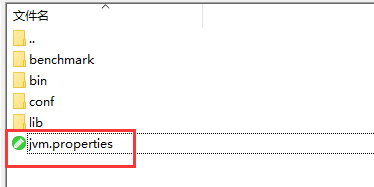
用户链接Amoeba的用户名和密码



定义读写策略



### 修改JVM内存

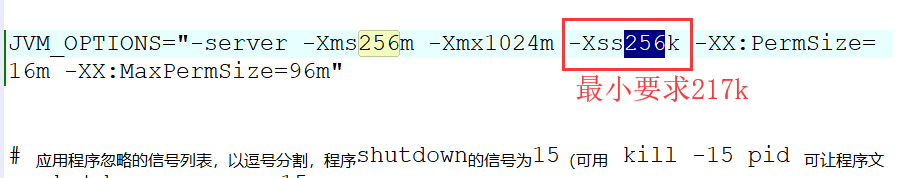


JVM\_OPTIONS="-server -Xms256m -Xmx1024m -Xss196k

-xms JVM初始化内存大小

-Xmx 最大内存大小

-Xss 线程内存大小

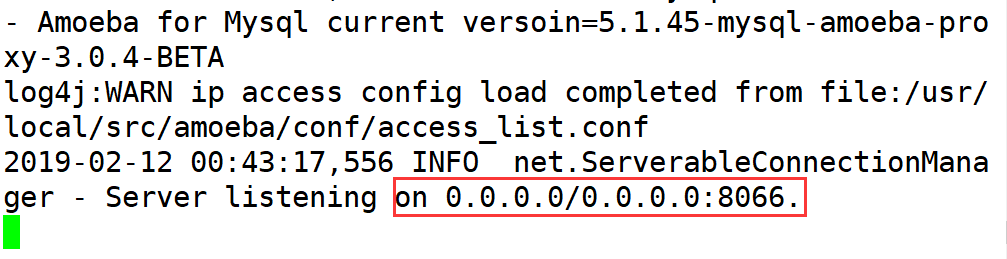


### 启动Amoeba

命令:

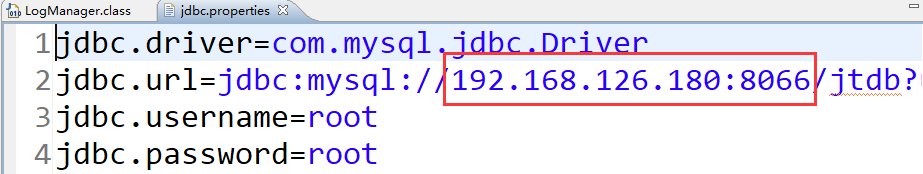
./launcher

./shutdown



### Amoeba读写分离/负载均衡测试

修改本机项目:链接远程amoeba服务.



1. 查询数据是否正常.
2. 修改数据之和检查从库是否同步数据.
3. 修改从库中title中数据.检查负载均衡是否正确

**修改完成之后,切记将数据修改相同!!!**