

Centos7上Mycat安装

1 环境准备

1. JDK1.7+ 此处使用的是 jdk-8u162-linux-x64.rpm
2. MySQL mysql-5.7.24-1.el7.x86_64.rpm-bundle.tar
下载镜像：http://mirrors.163.com/mysql/Downloads/MySQL-5.7/mysql-5.7.24-1.el7.x86_64.rpm-bundle.tar
3. Mycat 下载地址：<http://dl.mycat.io> 当前最新版本：1.6.6.1

2 JDK安装

1. 上传jdk安装包到linux机器上
2. root用户身份安装，命令：

```
rpm -ivh jdk-8u162-linux-x64.rpm
```

3. jdk被安装在 /usr/java目录下

```
[root@localhost ~]# ls /usr/java
default  jdk1.8.0_162  latest
```

4. 配置JAVA环境变量，命令：

```
vi /etc/profile
```

在文件末尾追加：

```
export JAVA_HOME=/usr/java/latest
export CLASSPATH=.:$JAVA_HOME/lib
export PATH=$JAVA_HOME/bin:$PATH
```

保存退出

使配置生效，命令：

```
source /etc/profile
```

5. 验证

```
java -version
```

```
[root@localhost ~]# java -version
java version "1.8.0_162"
Java(TM) SE Runtime Environment (build 1.8.0_162-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.162-b12, mixed mode)
```

3 MySQL安装

1. 上传MySQL安装包；
2. 在上传目录创建目录 mysql

```
mkdir mysql
```

```
[root@localhost ~]# ls
anaconda-ks.cfg
jdk-8u162-linux-x64.rpm
Mycat-server-1.6.6.1-release-20181031195535-linux.tar.gz
mysql-5.7.24-1.el7.x86_64.rpm-bundle.tar
[root@localhost ~]# mkdir mysql
```

3. 解压安装包

```
tar -xvf mysql-5.7.24-1.el7.x86_64.rpm-bundle.tar -C mysql
```

```
[root@localhost ~]#
[root@localhost ~]# tar -xvf mysql-5.7.24-1.el7.x86_64.rpm-bundle.tar -C mysql
mysql-community-common-5.7.24-1.el7.x86_64.rpm
mysql-community-minimal-debuginfo-5.7.24-1.el7.x86_64.rpm
mysql-community-embedded-compat-5.7.24-1.el7.x86_64.rpm
mysql-community-embedded-devel-5.7.24-1.el7.x86_64.rpm
mysql-community-embedded-5.7.24-1.el7.x86_64.rpm
mysql-community-libs-5.7.24-1.el7.x86_64.rpm
mysql-community-devel-5.7.24-1.el7.x86_64.rpm
mysql-community-server-5.7.24-1.el7.x86_64.rpm
mysql-community-libs-compat-5.7.24-1.el7.x86_64.rpm
mysql-community-client-5.7.24-1.el7.x86_64.rpm
mysql-community-server-minimal-5.7.24-1.el7.x86_64.rpm
mysql-community-test-5.7.24-1.el7.x86_64.rpm
[root@localhost ~]#
```

4. 安装：

```
cd mysql
rm -f mysql-community-server-minimal-5.7.24-1.el7.x86_64.rpm
yum install mysql-community-{server,client,common,libs}-* mysql-5.*
```

5. 安装完成，安装过程中会创建用户 mysql，所属组 mysql。mysql 内容的目录分布情况：

Files or Resources	Location
Client programs and scripts	/usr/bin
mysqld server	/usr/sbin
Configuration file	/etc/my.cnf
Data directory	/var/lib/mysql
Error log file	For RHEL, Oracle Linux, CentOS or Fedora platforms: /var/log/mysqld.log For SLES: /var/log/mysql/mysqld.log
Value of secure_file_priv	/var/lib/mysql-files
System V init script	For RHEL, Oracle Linux, CentOS or Fedora platforms: /etc/init.d/mysqld For SLES: /etc/init.d/mysql
Systemd service	For RHEL, Oracle Linux, CentOS or Fedora platforms: mysqld For SLES: mysql
Pid file	/var/run/mysql/mysqld.pid
Socket	/var/lib/mysql/mysql.sock
Keyring directory	/var/lib/mysql-keyring
Unix manual pages	/usr/share/man
Include (header) files	/usr/include/mysql
Libraries	/usr/lib/mysql
Miscellaneous support files (for example, error messages, and character set files)	/usr/share/mysql

6. 注意：Linux 下部署安装MySQL，默认不忽略表名大小写，需要手动到/etc/my.cnf 下配置 lower_case_table_names=1 使Linux 环境下MySQL 忽略表名大小写，否则使用MyCAT 的时候会提示找不到表的错误！

```
vi /etc/my.cnf
```

```
log-error=/var/log/mysql.log
pid-file=/var/run/mysql/mysql.pid
lower_case_table_names=1 ← 追加它
-- INSERT --
```

7. 启动mysql服务，命令：

```
systemctl start mysqld
```

第一次启动过程将完成如下事项：

- 服务器被初始化。
- 在数据目录中生成SSL证书和密钥文件。
- validate_password已安装并启用。
- 超级用户帐户'root'@'localhost'被创建。超级用户的密码被设置并存储在错误日志文件中。要提取它，请对RHEL、Oracle Linux、CentOS和Fedora系统使用以下命令：

```
grep 'temporary password' /var/log/mysql.log
```

```
[root@localhost mysql]#
[root@localhost mysql]# grep 'temporary password' /var/log/mysql.log
2019-02-22T10:03:16.341230Z 1 [Note] A temporary password is generated for root@localhost: CgUhV-P
<a5*h
[root@localhost mysql]#
```

8. 修改root用户密码，用上面提取到的密码登录：

```
mysql -uroot -p
```

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'MyNewPass4!';
```

9. 创建使用用户

```
mysql> grant all privileges on *.* to 'mike'@'%' identified by
'Mike666!';
```

```
mysql> flush privileges;
```

10. 安装完成，可以用我们的使用用户远程连接了！

4 Mycat安装

1. 解压安装包到目标安装目录：

```
tar -zxvf Mycat-server-1.6.6.1-release-20181031195535-linux.tar.gz -C /usr/local/
```

mycat软件目录构成：

```
[root@localhost ~]# ls /usr/local/mycat  
bin  catlet  conf  lib  logs  version.txt
```

- bin 程序目录，存放了window 版本和linux 版本，除了提供封装成服务的版本之外，也提供了nowrap 的 shell 脚本命令，方便大家选择和修改：

```
[root@localhost ~]# ls /usr/local/mycat/bin  
dataMigrate.sh  mycat  startup_nowrap.sh  wrapper-linux-x86-32  
init_zk_data.sh  rehash.sh  wrapper-linux-ppc-64  wrapper-linux-x86-64
```

- conf 目录下存放配置文件，server.xml 是Mycat 服务器参数调整和用户授权的配置文件，schema.xml 是逻辑库定义和表以及分片定义的配置文件，rule.xml 是分片规则的配置文件，分片规则的具体一些参数信息单独存放为文件，也在这个目录下，配置文件修改，需要重启Mycat 或者通过9066 端口reload。

```
[root@localhost ~]# ls /usr/local/mycat/conf  
autopartition-long.txt  migrateTables.properties  sequence_distributed_conf.properties  
auto-sharding-long.txt  myid.properties          sequence_time_conf.properties  
auto-sharding-rang-mod.txt  partition-hash-int.txt    server.xml  
cacheservice.properties  partition-range-mod.txt  sharding-by-enum.txt  
dbseq.sql                rule.xml                  wrapper.conf  
ehcache.xml              schema.xml                zkconf  
index_to_charset.properties  sequence_conf.properties  zkdownload  
log4j2.xml               sequence_db_conf.properties  
[root@localhost ~]#
```

- lib 目录下主要存放mycat 依赖的一些jar 文件。
- 日志存放在logs/mycat.log 中，每天一个文件，日志的配置是在conf/log4j2.xml 中，根据自己的需要，可以调整输出级别为debug，debug 级别下，会输出更多的信息，方便排查问题。

2. 配置环境变量MYCAT_HOME=/usr/local/mycat

```
vi /etc/profile
```

```
export JAVA_HOME=/usr/java/latest
export CLASSPATH=.:$JAVA_HOME/lib
export PATH=$JAVA_HOME/bin:$PATH

export MYCAT_HOME=/usr/local/mycat ← 追加这行
-- INSERT --
```

保存后，使其生效：

```
source /etc/profile
```

3. 新增用户mycat（建议不要用root用户来运行mycat）：

```
useradd mycat
```

设置密码

```
passwd mycat
```

4. 修改mycat安装目录的所有者为mycat用户：

```
chown -R mycat:mycat /usr/local/mycat
```

5. 尝试启动mycat服务

```
[root@localhost ~]#
[root@localhost ~]# su mycat ←
[mycat@localhost root]$ cd /usr/local/mycat/bin ←
[mycat@localhost bin]$ ls
dataMigrate.sh  mycat  startup_nowrap.sh  wrapper-linux-x86-32
init_zk_data.sh  rehash.sh  wrapper-linux-ppc-64  wrapper-linux-x86-64
[mycat@localhost bin]$ ./mycat ←
Usage: ./mycat { console | start | stop | restart | status | dump }
[mycat@localhost bin]$
```

```
./mycat start
```

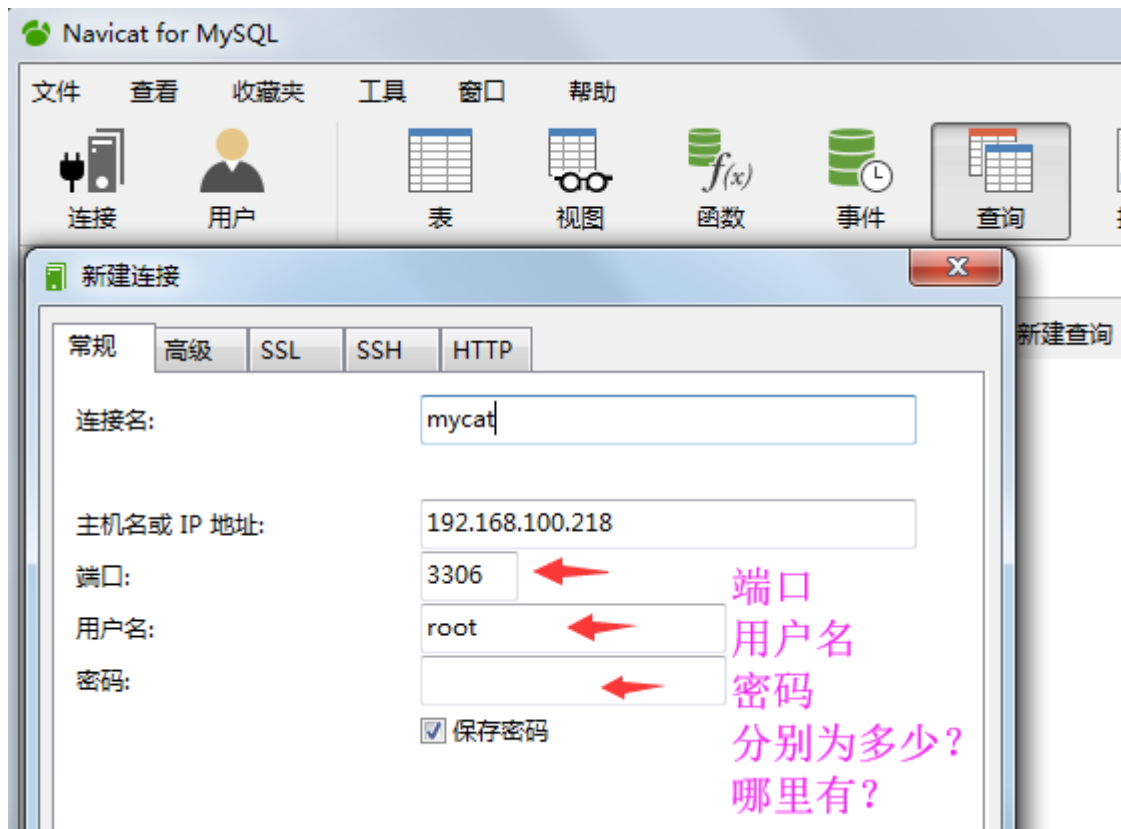
6. 启动成功否？看看有没有mycat进程

```

[mycat@localhost bin]$
[mycat@localhost bin]$ ./mycat start
Starting Mycat-server...
[mycat@localhost bin]$ jps
2549 Jps
2534 WrapperSimpleApp
[mycat@localhost bin]$ ps -ef | grep Mycat
mycat      2534    2532  11 21:23 ?        00:00:02 java -DMYCAT_HOME=. -server
:+AggressiveOpts -XX:MaxDirectMemorySize=2G -Dcom.sun.management.jmxremote

```

7. 用mysql客户端尝试连接mycat服务：



到 conf/server.xml 中看看：

```

<!--
<property name="serverPort">8066</property> <property name="managerPort">9066</property>
<property name="idleTimeout">300000</property> <property name="bindIp">0.0.0.0</property>
<property name="frontWriteQueueSize">4096</property> <property name="processors">32</property> -->

```

默认的服务端口为8066，管理端口为9066

用户有：

```

<user name="root" defaultAccount="true">
  <property name="password">123456</property>
  <property name="schemas">TESTDB</property>

  <!-- 表级 DML 权限设置 -->
  <!--
  <privileges check="false">
    <schema name="TESTDB" dml="0110" >
      <table name="tb01" dml="0000"></table>
      <table name="tb02" dml="1111"></table>
    </schema>
  </privileges>
  -->
</user>

<user name="user">
  <property name="password">user</property>
  <property name="schemas">TESTDB</property>
  <property name="readOnly">true</property>
</user>

```

看看用它们能连上不！

新建连接

常规 高级 SSL SSH HTTP

连接名: mycat

主机名或 IP 地址: 192.168.100.218

端口: 8066

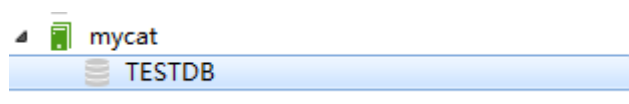
用户名: root

密码: ●●●●●●

☒ 保存密码

连接测试 确定 取消

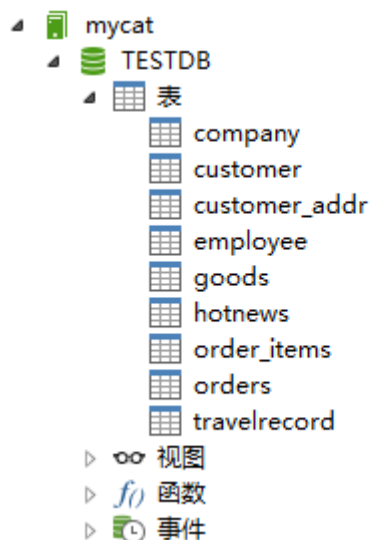
连接成功！



Mycat 对我们的应用来说，就是一个数据库！

疑问：TESTDB数据库怎么来的？

打开TESTDB数据库看看有什么表没！



疑问：表在哪里定义的？

看看表中有数据没！



不合法的数据源：0。

疑问：这里的数据源是什么？

这些疑问的答案从哪找？

conf/schema.xml

```
<?xml version="1.0"?>
<!DOCTYPE mycat:schema SYSTEM "schema.dtd">
<mycat:schema xmlns:mycat="http://io.mycat/">

    <schema name="TESTDB" checkSQLSchema="false" sqlMaxLimit="100">
```

```

    <!-- auto sharding by id (long) -->
    <table name="travelrecord" dataNode="dn1,dn2,dn3" rule="auto-
sharding-long" />

    <!-- global table is auto cloned to all defined data nodes ,so
can join
        with any table whose sharding node is in the same data
node -->
    <table name="company" primaryKey="ID" type="global"
dataNode="dn1,dn2,dn3" />
    <table name="goods" primaryKey="ID" type="global"
dataNode="dn1,dn2" />
    <!-- random sharding using mod sharind rule -->
    <table name="hotnews" primaryKey="ID" autoIncrement="true"
dataNode="dn1,dn2,dn3"
        rule="mod-long" />
    <table name="employee" primaryKey="ID" dataNode="dn1,dn2"
        rule="sharding-by-intfile" />
    <table name="customer" primaryKey="ID" dataNode="dn1,dn2"
        rule="sharding-by-intfile">
        <childTable name="orders" primaryKey="ID"
joinKey="customer_id"
            parentKey="id">
            <childTable name="order_items" joinKey="order_id"
                parentKey="id" />
            </childTable>
        <childTable name="customer_addr" primaryKey="ID"
joinKey="customer_id"
            parentKey="id" />
        </table>
</schema>

<dataNode name="dn1" dataHost="localhost1" database="db1" />
<dataNode name="dn2" dataHost="localhost1" database="db2" />
<dataNode name="dn3" dataHost="localhost1" database="db3" />

<dataHost name="localhost1" maxCon="1000" minCon="10" balance="0"
    writeType="0" dbType="mysql" dbDriver="native"
switchType="1" slaveThreshold="100">
    <heartbeat>select user()</heartbeat>
    <!-- can have multi write hosts -->
    <writeHost host="hostM1" url="localhost:3306" user="root"

```

```
        password="123456">
        <!-- can have multi read hosts -->
        <readHost host="hostS2" url="192.168.1.200:3306"
user="root" password="xxx" />
    </writeHost>
    <writeHost host="hostS1" url="localhost:3316" user="root"
        password="123456" />
</dataHost>
</mycat:schema>
```

8. 安装完成！