# 前置规则：

为方便软件的管理维护

将所有基础软件安装包均放置在/home/software 文件夹之下

mkdir /home/software

将所有软件均安装在 /usr/local 文件夹之下，以软件名作为包即可

# 基础软件

## Jdk-1.8

### 下载JDK

进入[jdk网站](https://links.jianshu.com/go?to=https%3A%2F%2Fwww.oracle.com%2Ftechnetwork%2Fjava%2Fjavase%2Fdownloads%2Fjdk8-downloads-2133151.html)，下载自己需要的版本

注意：这里需要下载 Linux 版本。本次安装以jdk-8u202-linux-x64.tar.gz为例。

把下载的jdk文件上传放在至 /home/software目录下

安装部署包中含有jdk-8u202-linux-x64.tar.gz 可直接拷贝

### 解压安装jdk文件

tar -zxvf /home/software/jdk-8u202-linux-x64.tar.gz

然后将解压后的文件，移动到指定目录，这个目录不是固定的，可以自定义，配置环境变量的时候，指定好实际路径就可以！

通常会将解压后的文件放在/usr/local/ jdk-8u202下：

sudo mv jdk-8u202 /usr/local/jdk-8u202

sudo cp /usr/local/jdk-8u202 /opt/jdk-8u202

### 设置环境变量(修改/etc/profile)

vi /etc/profile

在 profile 文件末尾添加如下内容并保存

export JAVA\_HOME=/opt/ jdk-8u202

export JRE\_HOME=${JAVA\_HOME}/jre

export CLASSPATH=.:${JAVA\_HOME}/lib:${JRE\_HOME}/lib:$CLASSPATH

export JAVA\_PATH=${JAVA\_HOME}/bin:${JRE\_HOME}/bin

export PATH=$PATH:${JAVA\_PATH}

编辑完成之后，按ESC键 跳到命令模式（命令前面冒号是必要的）:wq

扩展：

* :w 保存文件但不退出vi
* :w file 将修改另外保存到file中，不退出vi
* :w! 强制保存，不推出vi
* :wq 保存文件并退出vi
* :wq! 强制保存文件，并退出vi
* :q 不保存文件，退出vi
* :q! 不保存文件，强制退出vi
* :e! 放弃所有修改，从上次保存文件开始再编辑

### 测试安装

java -version

显示你所安装的java的版本信息，则说明jdk安装成功

[root@testserver-120 software]# java -version

openjdk version "1.8.0\_262"

OpenJDK Runtime Environment (build 1.8.0\_262-b10)

OpenJDK 64-Bit Server VM (build 25.262-b10, mixed mode)

## Mysql

### 首先卸载Centos7自带的Mariadb

# 查看系统自带的Mariadb

rpm -qa|grep mariadb

mariadb-libs-5.5.44-2.el7.centos.x86\_64

# 卸载系统自带的Mariadb(此时要根据实际情况去卸载)

rpm -e --nodeps mariadb-libs-5.5.56-2.el7.x86\_64

### 查看用户和组是否存在

#### 先检查是否存在，如果不存在则创建

# 检查mysql组和用户是否存在，如无则创建

cat /etc/group | grep mysql

cat /etc/passwd | grep mysql

# 创建mysql用户组

groupadd mysql

# 创建一个用户名为mysql的用户，并加入mysql用户组

useradd -g mysql mysql

# 制定password 为111111

passwd mysql

Changing password for user mysql.

New password:

BAD PASSWORD: The password is a palindrome

Retype new password:

passwd: all authentication tokens updated successfully.

### 下载mysql离线安装包tar文件

官网下载地址：[https://dev.mysql.com/downloa...](https://link.segmentfault.com/?enc=DJpD50WDlK%2B91Xyk3NU3Jg%3D%3D.ndnRVD9P%2F%2FZa6%2B3KzRykFbKX6v3h%2BRKYLMJ4rV1qZJcVDTUcfiJPDkE8MpOputuLeBo%2Bh7y43lyrYTF5CKUCHQ%3D%3D)

版本选择，可以选择一下两种方式：

1）使用Red Hat Enterprise Linux  
Select Version:**5.7.25**  
Select Operating **System:Red Hat Enterprise Linux / Oracle Linux**  
Select OS Version:**Red Hat Enterprise Linux 7 / Oracle Linux 7 (x86, 64-bit)**  
列表中下载：  
Compressed TAR Archive：(mysql-5.7.9-el7-x86\_64.tar.gz)

2）使用Linux - Generic  
Select Version:**5.7.25**  
Select Operating System:**Linux - Generic**  
Select OS Version:**Linux - Generic (glibc 2.12) (x86, 64-bit)**  
列表中下载：  
Compressed TAR Archive：(mysql-5.7.9-linux-glibc2.5-x86\_64.tar.gz)【本文中使用的是这个版本】  
注意：上边两种方式找mysql离线安装包的方式都可以。

本次使用：mysql-5.7.9-linux-glibc2.5-x86\_64.tar.gz

部署包中带有 mysql-5.7.9-linux-glibc2.5-x86\_64.tar.gz 的安装包

### 上传第四步下载的mysql TAR包

# 上传mysql TAR包

# 或者直接使用XFTP进行安装

rz

# 解压mysql-5.7.9-linux-glibc2.5-x86\_64.tar.gz

# ls

bin full-path-to-mysql-VERSION-OS include lib64 mysql-5.7.9-linux-glibc2.5-x86\_64.tar.gz share

etc games lib libexec sbin src

# tar -zxvf mysql-5.7.9-linux-glibc2.5-x86\_64.tar.gz -C /usr/local/mysql

...

mysql-5.7.9-linux-glibc2.12-x86\_64/share/install\_rewriter.sql

mysql-5.7.9-linux-glibc2.12-x86\_64/share/uninstall\_rewriter.sql

mysql-5.7.9-linux-glibc2.12-x86\_64/support-files/magic

mysql-5.7.9-linux-glibc2.12-x86\_64/support-files/mysql.server

mysql-5.7.9-linux-glibc2.12-x86\_64/docs/INFO\_BIN

mysql-5.7.9-linux-glibc2.12-x86\_64/docs/INFO\_SRC

[root@localhost ~]# cd /usr/local/

[root@localhost ~]# ls

bin etc full-path-to-mysql-VERSION-OS games include lib lib64 libexec mysql mysql-5.7.9-linux-glibc2.5-x86\_64.tar.gz sbin share src

### 在/etc下创建my.cnf文件

# 编辑/etc/my.cnf

[root@CDH-141 mysql]# vi /etc/my.cnf

输入以下代码

[mysql]

socket=/tmp/mysql.sock

# set mysql client default chararter

default-character-set=utf8

[mysqld]

socket=/tmp/mysql.sock

# set mysql server port

port = 3306 # 默认是3306，如果这里发现3306已经被占用，可以更改

# set mysql install base dir

basedir=/usr/local/mysql

# set the data store dir

datadir=/usr/local/mysql/data

# set the number of allow max connnection

max\_connections=1024

# set server charactre default encoding

character-set-server=utf8

# the storage engine

default-storage-engine=INNODB

# 设置MySQL对表名等不区分大小写

lower\_case\_table\_names=1

max\_allowed\_packet=200M

explicit\_defaults\_for\_timestamp=true

#阻止过多尝试失败的客户端以防止暴力破解密码的情况,与性能并无太大的关系

max\_connect\_errors=30

#此参数确定数据日志文件的大小，以M为单位，根据数据更新频率调整。

innodb\_log\_file\_size=50

#指定大小的内存来缓冲数据和索引,最大可以把该值设置成物理内存的80%

innodb\_buffer\_pool\_size=10G

key\_buffer\_size=16M

sql\_mode=STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION

# 配置GROUP\_CONCAT拼接的字符串的长度字节

group\_concat\_max\_len = 102400

[mysql.server]

user=mysql

basedir=/usr/local/mysql

# 可以防止出现插入中文报错；如果此时不生效，可以强制在创建表的时候指定使用utf8的编码集

[client]

default-character-set = utf8

### 进入mysql文件夹，并安装mysql

# 进入mysql

# cd /usr/local/mysql

# 安装mysql

[root@CDH-141 mysql]# bin/mysql\_install\_db --user=mysql --basedir=/usr/local/mysql/ --datadir=/usr/local/mysql/data/

2019-03-08 18:11:07 [WARNING] mysql\_install\_db is deprecated. Please consider switching to mysqld --initialize

2019-03-08 18:11:24 [WARNING] The bootstrap log isn't empty:

2019-03-08 18:11:24 [WARNING] 2019-03-08T10:11:07.208602Z 0 [Warning] --bootstrap is deprecated. Please consider using --initialize instead

# 设置文件以及目录的权限

[root@CDH-141 mysql]# cp ./support-files/mysql.server /etc/init.d/mysqld

### 启动MySQL

# 启动mysql

[root@CDH-141 mysql]# /etc/init.d/mysqld restart

ERROR! MySQL server PID file could not be found!

Starting MySQL.Logging to '/usr/local/mysql/data/localhost.localdomain.err'.

SUCCESS!

### 设置开机自启动

#设置开机启动

[root@CDH-141 mysql]# chkconfig --level 35 mysqld on

[root@CDH-141 mysql]# chkconfig --list mysqld

Note: This output shows SysV services only and does not include native

systemd services. SysV configuration data might be overridden by native

systemd configuration.

If you want to list systemd services use 'systemctl list-unit-files'.

To see services enabled on particular target use

'systemctl list-dependencies [target]'.

mysqld 0:off 1:off 2:on 3:on 4:on 5:on 6:off

[root@CDH-141 mysql]# chkconfig --add mysqld

[root@CDH-141 mysql]# chkconfig --list mysqld

Note: This output shows SysV services only and does not include native

systemd services. SysV configuration data might be overridden by native

systemd configuration.

If you want to list systemd services use 'systemctl list-unit-files'.

To see services enabled on particular target use

'systemctl list-dependencies [target]'.

mysqld 0:off 1:off 2:on 3:on 4:on 5:on 6:off

[root@CDH-141 mysql]# service mysqld status

MySQL running (26122)[ OK ]

[root@CDH-141 mysql]#

### 修改环境变量配置文件

# 进入/etc/profile文件夹

[root@CDH-141 mysql]# vi /etc/profile

修改/etc/profile，在最后添加如下内容

# 修改/etc/profile文件

export MYSQL\_HOME=/usr/local/mysql

export PATH=$MYSQL\_HOME/bin:$PATH

# 使文件生效

[root@CDH-141 mysql]# source /etc/profile

### 获得MySQL初始密码

# 每个人的密码是不一样的，随机生成的

[root@CDH-141 mysql]# cat /root/.mysql\_secret

# Password set for user 'root@localhost' at 2019-03-08 17:40:42

poc3u0mO\_luv

[root@CDH-141 mysql]#

### 修改密码

[root@CDH-141 mysql]# mysql -uroot -p

Enter password: #此处填写上边获取到的初始密码‘poc3u0mO\_luv’

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 2

Server version: 5.7.25

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> set PASSWORD = PASSWORD('111111');

Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> flush privileges;

Query OK, 0 rows affected (0.00 sec)

mysql> exit

Bye

# 验证登录

[root@localhost mysql]# mysql -uroot -p111111

mysql: [Warning] Using a password on the command line interface can be insecure.

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 5

Server version: 5.7.27 MySQL Community Server (GPL)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its

affiliates. Other names may be trademarks of their respective

owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

### 添加远程访问权限

注意：当前的访问权限是直接给root用户添加的，如果在项目现场部署的时候，是不能使用root用户的;

此时是有两种方案的，分别做了整理：

## 方案1：直接给root用户增加远程访问权限

# 添加远程访问权限

mysql> use mysql

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

mysql> update user set host='%' where user='root';

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> select host,user from user;

+-----------+---------------+

| host | user |

+-----------+---------------+

| % | root |

| localhost | mysql.session |

| localhost | mysql.sys |

+-----------+---------------+

3 rows in set (0.00 sec)

mysql>

--- 方案2：追加一个demo用户，然后给demo用户添加远程访问的权限

mysql> use mysql

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

-- 此时是给所有用户开放的，如果需要对不不同的用户开发需要再单独设置

mysql> update user set host='%' where user='demo';

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> select host,user from user;

+-----------+---------------+

| host | user |

+-----------+---------------+

| % | demo |

| localhost | mysql.session |

| localhost | mysql.sys |

+-----------+---------------+

3 rows in set (0.00 sec)

### 重启MySQL使其生效

# 重启mysql

[root@CDH-141 mysql]# /etc/init.d/mysqld restart

Shutting down MySQL..[ OK ]

Starting MySQL..[ OK ]

[root@CDH-141 mysql]#

## 重启命令2

systemctl start mysql

### 如果连接失败，需要检查防火墙

# 第一步：关闭防火墙

systemctl stop firewalld

#################

#############

####### 注意，在实际中先关闭，确定是使用开发端口的方式实现，还是直接移除，再执行对应的操作

##### 第二步：此时有两种解决方案

# 方案1：移除防火墙

systemctl disable firewalld

# 方案2：开发3306端口

# 开放端口请根据实际访问规则配置，这里只是单纯开启3306端口的访问，默认全放行，在生产环境禁止开放所有IP访问

# (1) 向防火墙添加 mysql 端口:

firewall-cmd --zone=public --add-port=3306/tcp --permanent

# (2) 刷新防火墙规则:

firewall-cmd --reload

# (3) 验证端口,查询防火墙开放端口:

firewall-cmd --zone=public --list-port

## Mycat

### 下载并解压

下载并上传 Linux 服务器中，进行解压，解压即安装

tar -zxvf mycat2.tar.gz

mv mycat2 /home

### 概念描述

#### 分库分表

按照一定规则把数据库中的表拆分为多个带有数据库实例,物理库,物理表访问路径的分表。

解读：分库：一个电商项目，分为用户库、订单库等等。

分表：一张订单表数据数百万，达到 MySQL 单表瓶颈，分到多个数据库中的多张表

#### 逻辑库

数据库代理中的数据库,它可以包含多个逻辑表。

解读：Mycat 里定义的库，在逻辑上存在，物理上在 MySQL 里并不存在。有可能是多个 MySQL 数据库共同组成一个逻辑库。类似多个小孩叠罗汉穿上外套，扮演一个大人。

#### 逻辑表

数据库代理中的表,它可以映射代理连接的数据库中的表(物理表)

解读：Mycat 里定义的表，在逻辑上存在，可以映射真实的 MySQL 数据库的表。可

以一对一，也可以一对多。

#### 物理库

数据库代理连接的数据库中的库

解读：MySQL 真实的数据库。

#### 物理表

数据库代理连接的数据库中的表

解读：MySQL 真实的数据库中的真实数据表。

#### 拆分键

即分片键,描述拆分逻辑表的数据规则的字段

解读：比如订单表可以按照归属的用户 id 拆分，用户 id 就是拆分键

#### 物理分表

指已经进行数据拆分的,在数据库上面的物理表,是分片表的一个分区

解读：多个物理分表里的数据汇总就是逻辑表的全部数据

#### 物理分库

一般指包含多个物理分表的库

解读：参与数据分片的实际数据库

#### 分库

一般指通过多个数据库拆分分片表,每个数据库一个物理分表,物理分库名字相同

解读：分库是个动作，需要多个数据库参与。就像多个数据库是多个盘子，分库就是

把一串数据葡萄，分到各个盘子里，而查询数据时，所有盘子的葡萄又通过 Mycat2 组

成了完整的一串葡萄。

#### 分片表,水平分片表

按照一定规则把数据拆分成多个分区的表,在分库分表语境下,它属于逻辑表的一种

解读：安按照规则拆分数据，上个例子中的那串葡萄。

#### 单表

没有分片,没有数据冗余的表,

解读：没有拆分数据，也没有复制数据到别的库的表。

#### 全局表,广播表

每个数据库实例都冗余全量数据的逻辑表.

它通过表数据冗余,使分片表的分区与该表的数据在同一个数据库实例里,达到 join 运

算能够直接在该数据库实例里执行.它的数据一致一般是通过数据库代理分发 SQL 实现.

也有基于集群日志的实现

解读：例如系统中翻译字段的字典表，每个分片表都需要完整的字典数据翻译字段。

#### ER 表

狭义指父子表中的子表,它的分片键指向父表的分片键,而且两表的分片算法相同

广义指具有相同数据分布的一组表

解读：关联别的表的子表，例如：订单详情表就是订单表的 ER 表

#### 集群

多个数据节点组成的逻辑节点.在 mycat2 里,它是把对多个数据源地址视为一个数据源

地址(名称),并提供自动故障恢复,转移,即实现高可用,负载均衡的组件。

解读：集群就是高可用、负载均衡的代名词

#### 数据源

连接后端数据库的组件,它是数据库代理中连接后端数据库的客户端

解读：Mycat 通过数据源连接 MySQL 数据库

#### 原型库(prototype)

原型库是 Mycat2 后面的数据库，比如 mysql 库

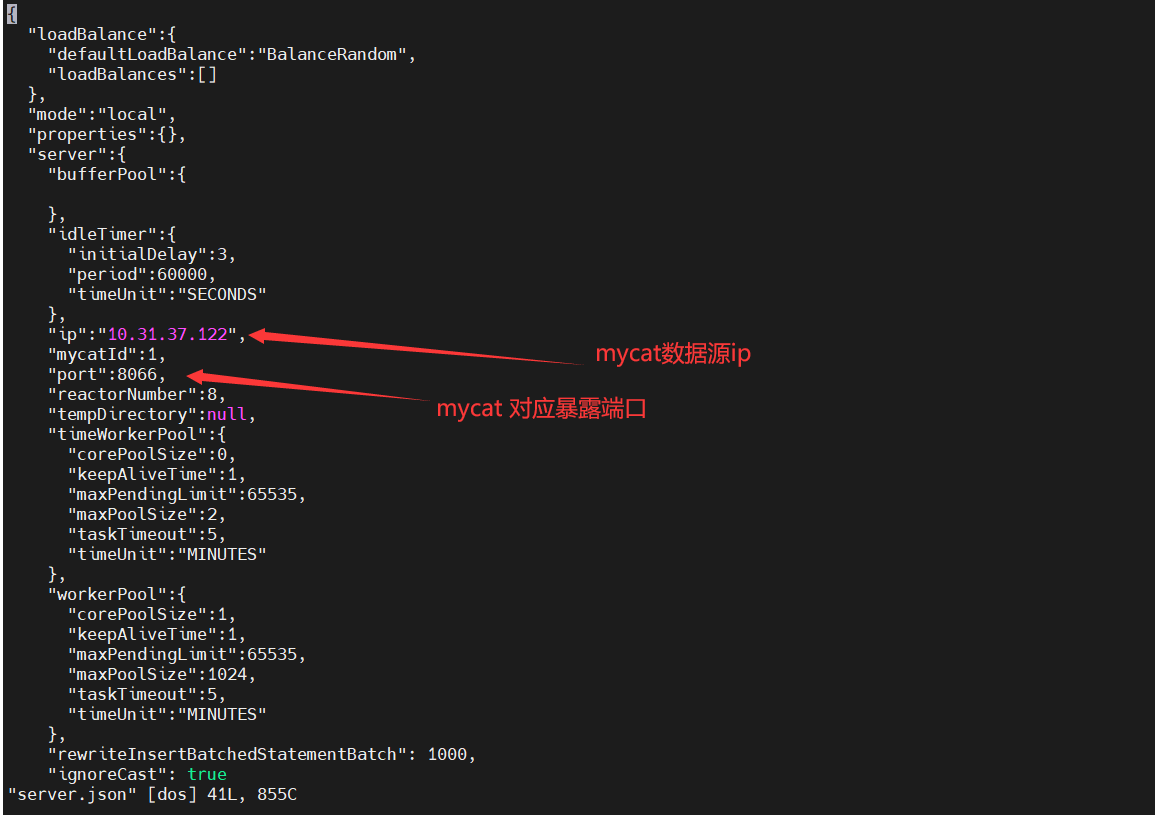
解读：原型库就是存储数据的真实数据库，配置数据源时必须指定原型库

### 修改对应配置

cd /home/mycat2/mycat/conf

#### 服务相关配置

vi /home/mycat2/mycat/conf/server.json



仅修改对应的mycat ip 端口即可

#### 数据源配置

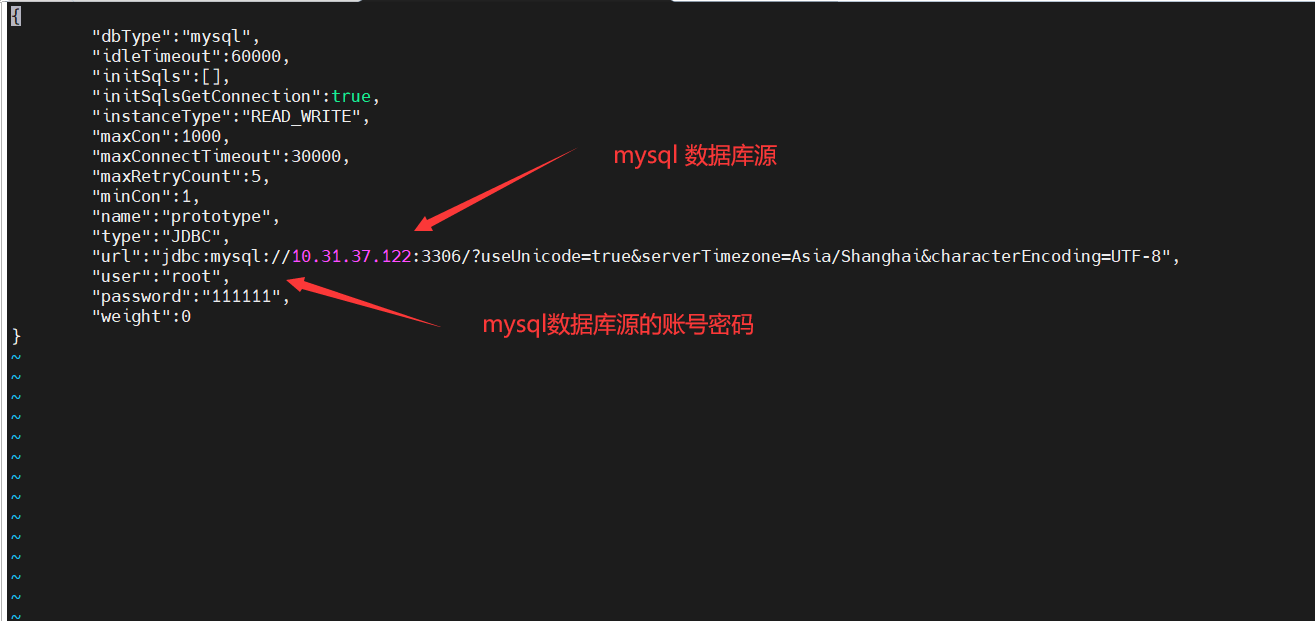
cd dataSources/

Vim /home/mycat2/mycat/conf/datasources/prototype.datasource.json

命名方式 ：

{数据源名字}.datasource.json

修改对应的数据库连接信息



#字段含义

# dbType：数据库类型，mysql

# name：用户名

# password：密码

# type：数据源类型，默认 JDBC

# url：访问数据库地址

# idleTimeout：空闲连接超时时间

# initSqls：初始化sql

# initSqlsGetConnection：对于 jdbc 每次获取连接是否都执行 initSqls

# instanceType：配置实例只读还是读写

可选值: READ\_WRITE, READ, WRITE

#weight ：负载均衡权重

# 连接相关配置

"maxCon": 1000, "maxConnectTimeout": 3000, "maxRetryCount": 5, "minCon": 1,

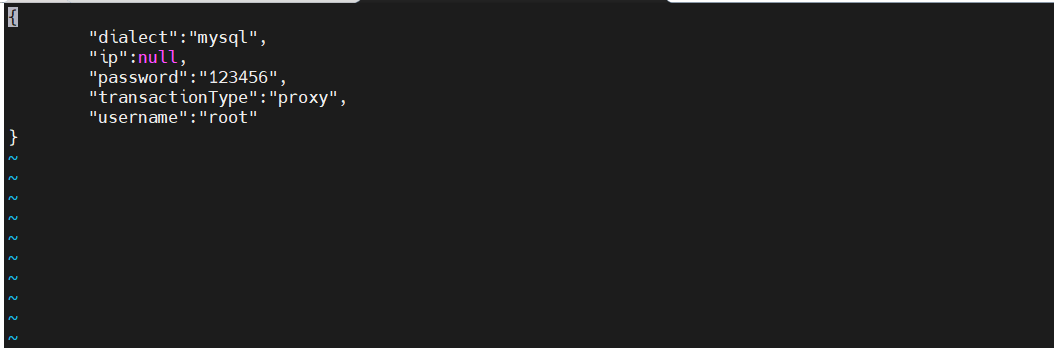
#### 配置用户相关信息

Cd /home/mycat2/mycat/conf/users

**命名方式**

{用户名}.user.json

vi /home/mycat2/mycat/conf/users/root.user.json



字段说明：

#字段含义

#ip：客户端访问ip，建议为空,填写后会对客户端的ip进行限制

# username：用户名

# password：密码

# transactionType：事务类型

可选值: proxy 本地事务,在涉及大于 1 个数据库的事务,commit 阶段失败会导致不一致,但是兼容性最好

xa 事务,需要确认存储节点集群类型是否支持 XA

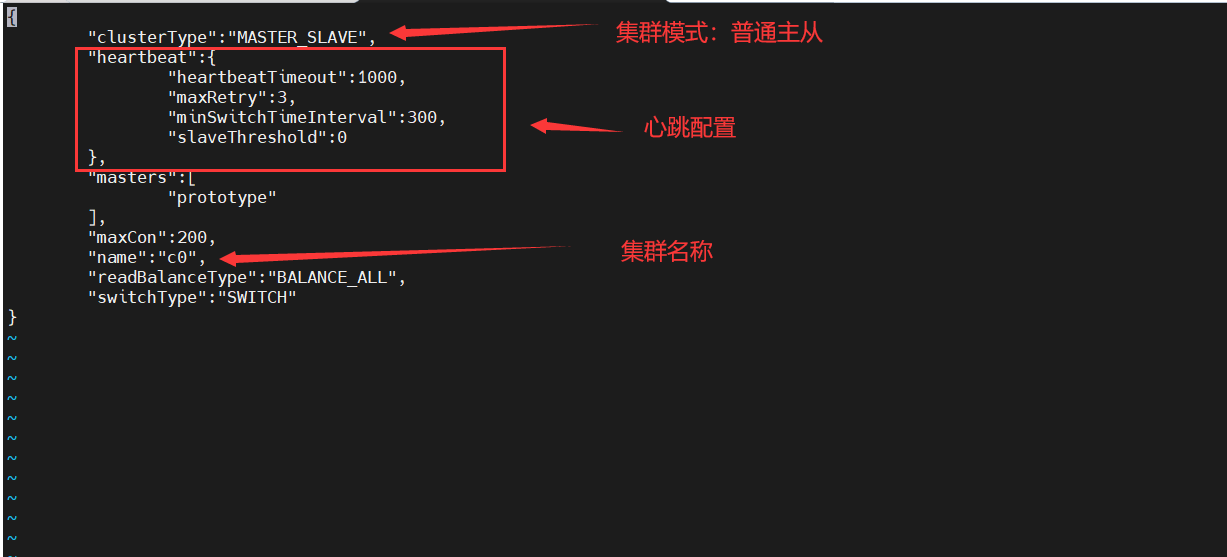
可以通过语句实现切换 set transaction\_policy = 'xa' set transaction\_policy = 'proxy'

可以通过语句查询 SELECT @@transaction\_policy

#### 集群配置信息

{集群名字}.cluster.json

Vi /home/mycat2/mycat/conf/clusters/c0.cluster.json



#字段含义

# clusterType：集群类型

可选值:

SINGLE\_NODE:单一节点

MASTER\_SLAVE: 普通主从

GARELA\_CLUSTER:garela cluster/PXC 集群

MHA：MHA 集群

MGR：MGR 集群

# readBalanceType：查询负载均衡策略

可选值:

BALANCE\_ALL(默认值) 获取集群中所有数据源

BALANCE\_ALL\_READ 获取集群中允许读的数据源

BALANCE\_READ\_WRITE 获取集群中允许读写的数据源,但允许读的数据源优先 BALANCE\_NONE 获取集群中允许写数据源,即主节点中选择

# switchType：切换类型

可选值:

NOT\_SWITCH:不进行主从切换

SWITCH:进行主从切换

#### 逻辑库表配置（schema）

Cd /home/mycat2/mycat/conf/schemas

命名方式

{库名}.schema.json

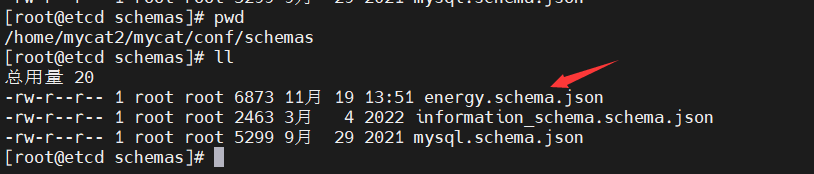
# 字段解释：

# schemaName：逻辑库名

# targetName：目的数据源或集群

配置实例：

将安装包中：部署\linux基础服务\mycat2dbsql\energy.schema.json 配置文件复制在/home/mycat2/mycat/conf/schemas中，进行覆盖



然后将mycat进行重启即可。

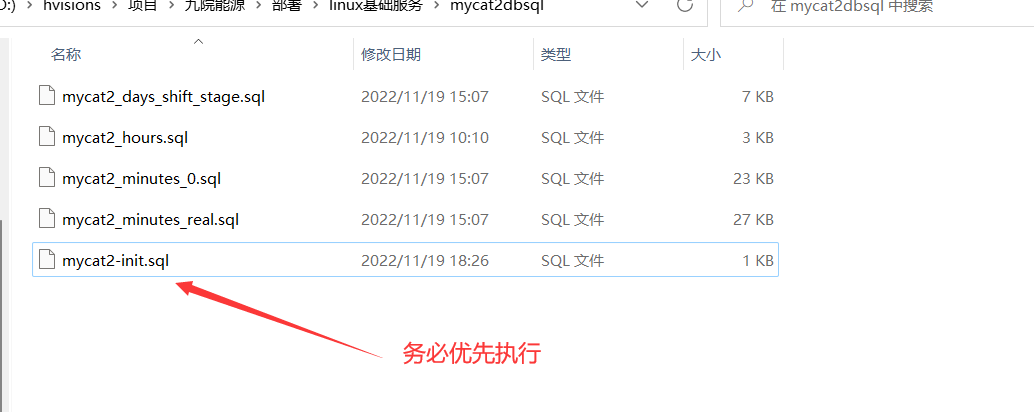
**对应配置解释**



### 初始化数据库

执行脚本 部署\linux基础服务\mycat2dbsql

优先执行： mycat2-init.sql



### 启动mycat2

Bash /home/mycat2/mycat/bin/mycat start

即可启动成功

### 测试连接

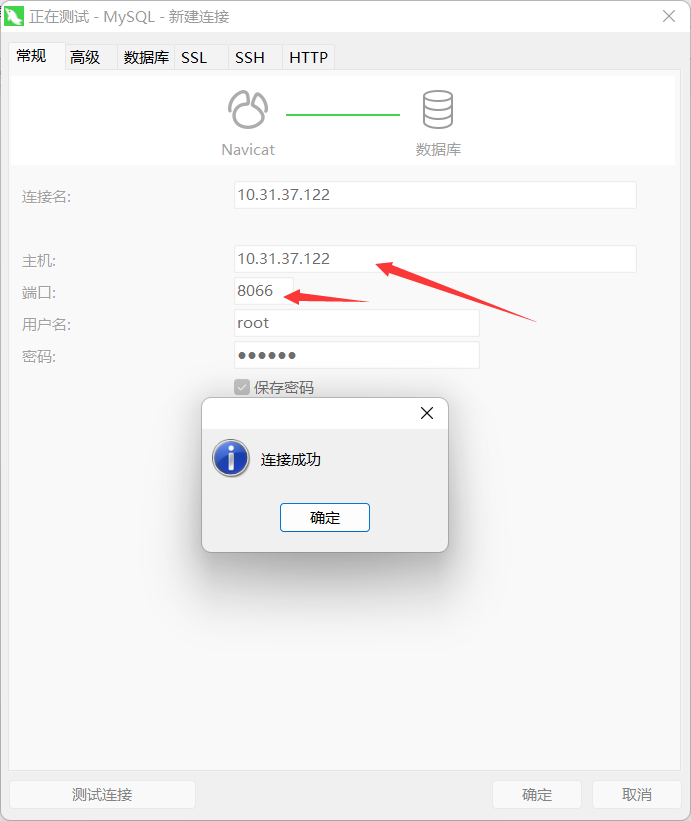
使用navicat 连接

Ip

端口： 8066

用户名: root

密码： 123456



## nginx

### 准备阶段

1. 查看 操作系统是否安装 gcc、gcc-c++；
2. 从 CentOS 7 镜像中，提取依赖安装包：gcc、gcc-c++；
3. 下载Nginx需要依赖的离线安装包：ssl、pcre、zlib；
4. 下载Nginx离线安装包：nginx-1.18.0.tar.gz。

Nginx 所需要的依赖较多，包括以下

cmake-2.8.12.2-2.el7.x86\_64.rpm make-3.82-24.el7.x86\_64.rpm gcc-4.8.5-39.el7.x86\_64.rpm gcc-c++-4.8.5-39.el7.x86\_64.rpm pcre-devel-8.32-17.el7.x86\_64.rpm pcre2-10.23-2.el7.x86\_64.rpm zlib-devel-1.2.7-18.el7.x86\_64.rpm openssl-devel-1.0.2k-8.el7.x86\_64.rpm

所有依赖包所在位置： \部署\linux基础服务\nginx-lib

将所有依赖包下载并上传至 /hom/software/nginx-lib 中

然后执行

rpm -Uvh \*.rpm –nodeps

若发生其他异常，则手动安装所有库

例如：

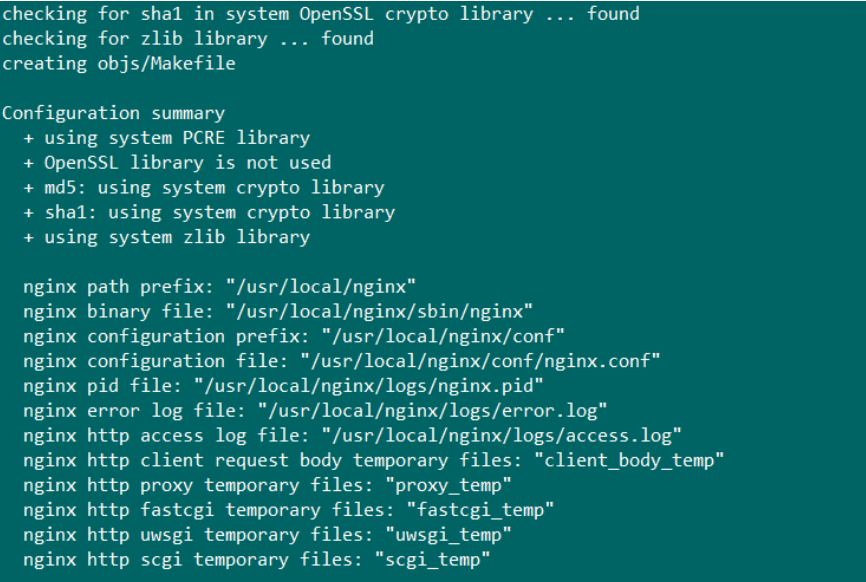
rpm -Uvh cmake-2.8.12.2-2.el7.x86\_64.rpm –nodeps

….

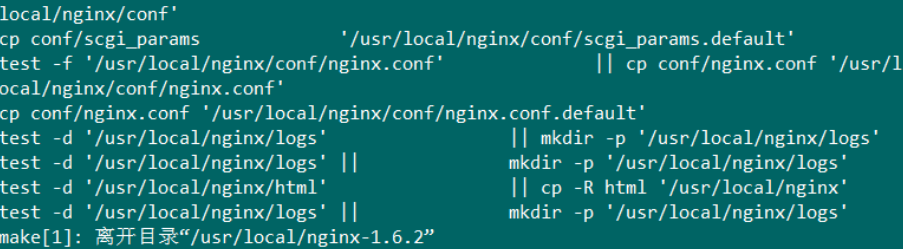
### 解压并安装

tar zxvf nginx-1.9.9.tar.gz -C /usr/local/

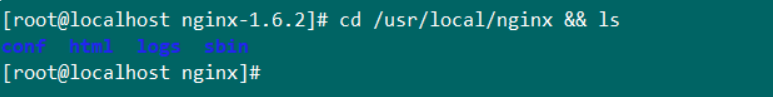
cd /usr/local/nginx-1.9.9/ && ./configure --prefix=/usr/local/nginx



make && make install



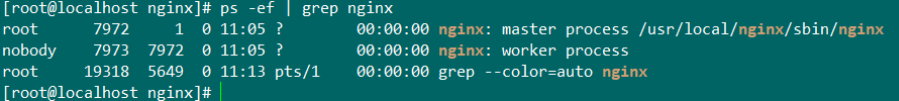
cd /usr/local/nginx && ls



### 启动nginx

/usr/local/nginx/sbin/nginx

ps -ef | grep nginx



### 防火墙端口通行

firewall-cmd --add-port=80/tcp --permanent

firewall-cmd --reload

firewall-cmd --list-ports

### web访问Nginx测试

打开网页： 访问 ip:80 端口即可



**nginx -s reload**

## docker

### 解压

tar -zxvf docker-20.10.18.tgz

### 将解压出来的docker文件复制到 /usr/bin/ 目录下

cp docker/\* /usr/bin/

### 进入/etc/systemd/system/目录,并创建docker.service文件

cd /etc/systemd/system/

touch docker.service

### 打开docker.service文件,将以下内容复制

vim docker.service

[Unit]

Description=Docker Application Container Engine

Documentation=https://docs.docker.com

After=network-online.target firewalld.service

Wants=network-online.target

[Service]

Type=notify

# the default is not to use systemd for cgroups because the delegate issues still

# exists and systemd currently does not support the cgroup feature set required

# for containers run by docker

ExecStart=/usr/bin/dockerd --selinux-enabled=false --insecure-registry=192.168.200.128

ExecReload=/bin/kill -s HUP $MAINPID

# Having non-zero Limit\*s causes performance problems due to accounting overhead

# in the kernel. We recommend using cgroups to do container-local accounting.

LimitNOFILE=infinity

LimitNPROC=infinity

LimitCORE=infinity

# Uncomment TasksMax if your systemd version supports it.

# Only systemd 226 and above support this version.

#TasksMax=infinity

TimeoutStartSec=0

# set delegate yes so that systemd does not reset the cgroups of docker containers

Delegate=yes

# kill only the docker process, not all processes in the cgroup

KillMode=process

# restart the docker process if it exits prematurely

Restart=on-failure

StartLimitBurst=3

StartLimitInterval=60s

[Install]

WantedBy=multi-user.target

### 给docker.service文件添加执行权限

chmod 777 /etc/systemd/system/docker.service

### 重新加载配置文件（每次有修改docker.service文件时都要重新加载下）

systemctl daemon-reload

### 启动

systemctl start docker

### 设置开机启动

systemctl enable docker.service

### 查看docker状态

systemctl status docker.service

出现下面这个界面就代表docker安装成功。

# 容器

## Redis

### 加载tar文件， 进行生成redis镜像

docker load -i redis.tar

### 镜像查看

docker images

### 运行redis镜像

docker run \

-p 6379:6379 \

-v /programs/redis/data:/data \

-v /programs/redis/conf/redis.conf:/etc/redis/redis.conf \

--privileged=true \

--name myredis \

-d \

redis redis-server /etc/redis/redis.conf

### 参数解读：

1. -p:指定宿主端口和容器暴露接口
2. --name：指定容器名称
3. -d：后台运行
4. --restart：是否重启
5. -v：容器文件挂载到宿主机上
6. redis ：容器名称
7. redis-server /etc/redis/redis.conf：指定配置文件启动

## Nacos

### 解压加载naocs 镜像

docker load -i nacos.tar

### 初始化nacos数据库

部署\linux镜像\init-sql-container\nacos-init.sql

### 启动nacos容器

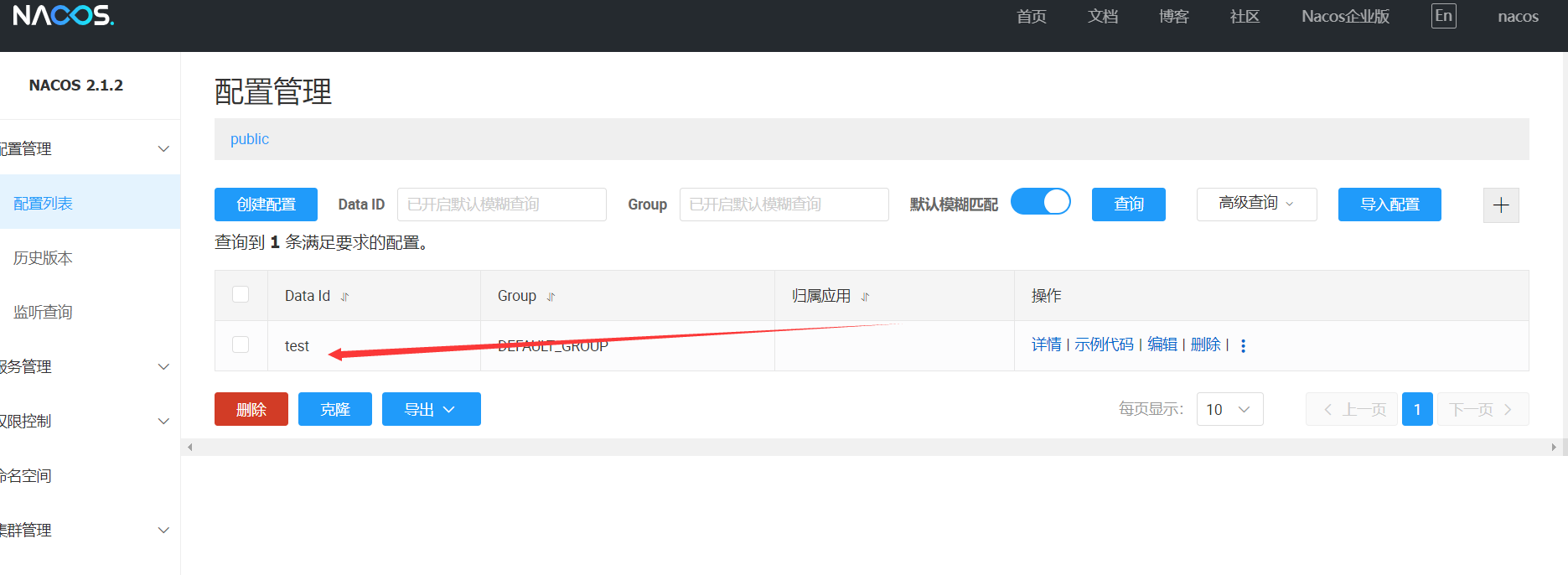
docker run -d -p 8848:8848 -p 9848:9848 -p 9849:9849 \ --restart=always \ --name nacos \ --env MODE=standalone \ --env SPRING\_DATASOURCE\_PLATFORM=mysql \ --env MYSQL\_SERVICE\_HOST=10.31.37.122 \ --env MYSQL\_SERVICE\_PORT=3306 \ --env MYSQL\_SERVICE\_DB\_NAME=nacos \ --env MYSQL\_SERVICE\_USER=root \ --env MYSQL\_SERVICE\_PASSWORD=111111 \ nacos/nacos-server:v2.1.2

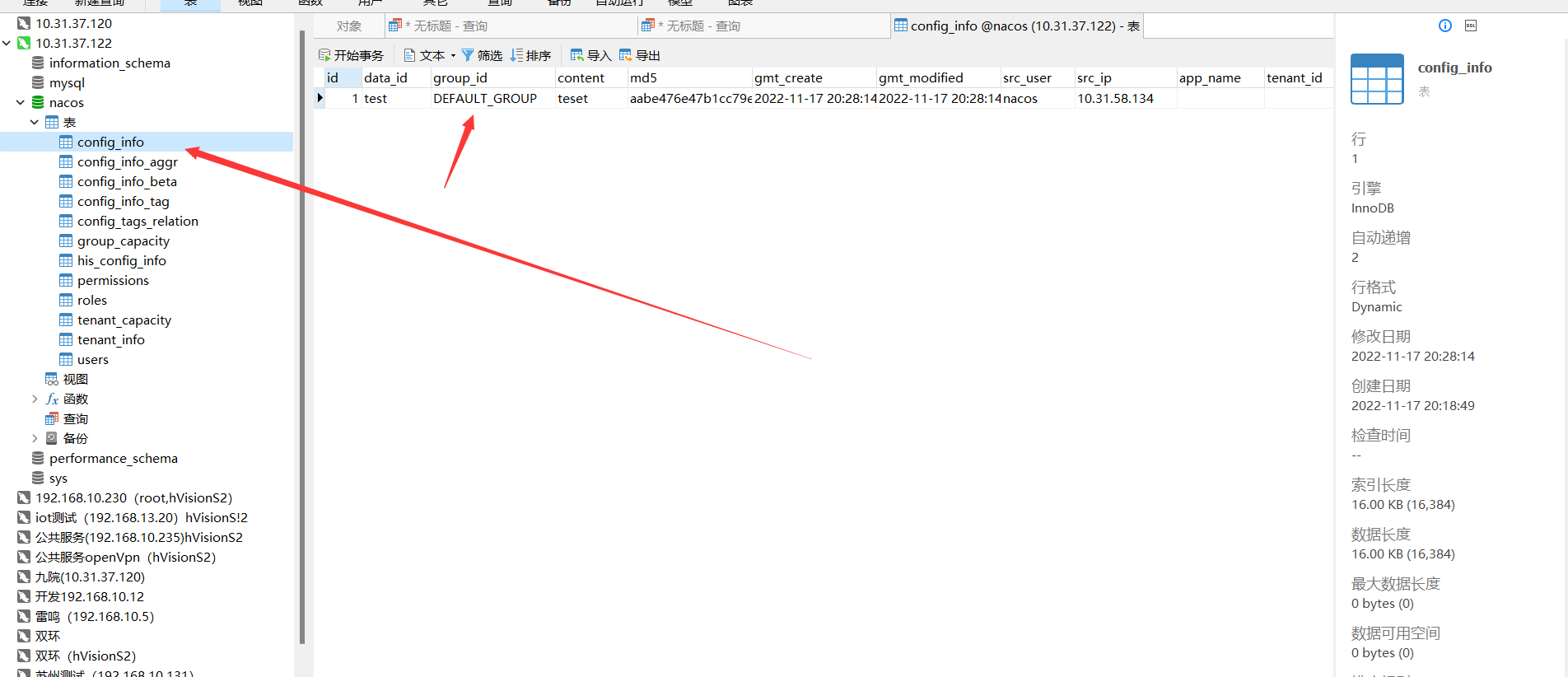
注意： nacos 2.0.2之后的版本，不止需要开放8848端口，还需要9848，9849两个端口。

### 测试

在浏览器上输入访问地址：http://10.31.37.122:8848/nacos，测试是否安装成功

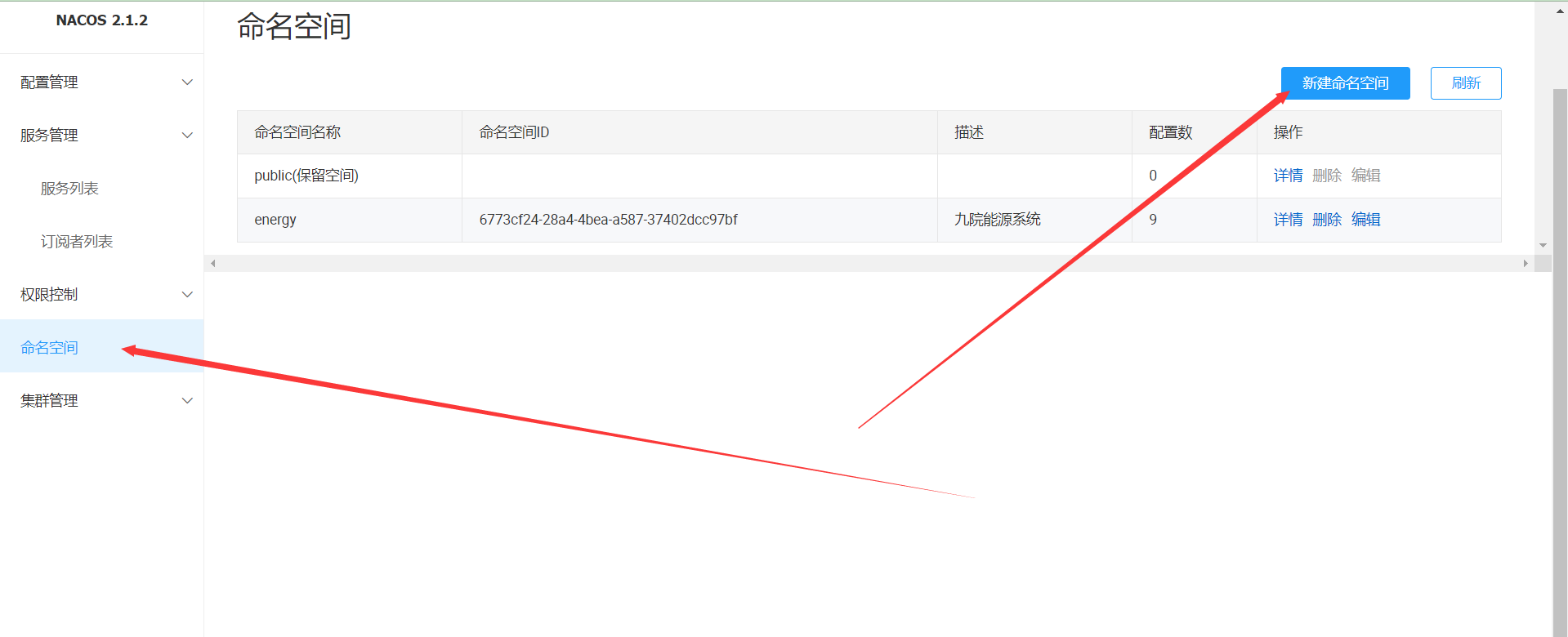
添加一条数据，看数据库是否记录

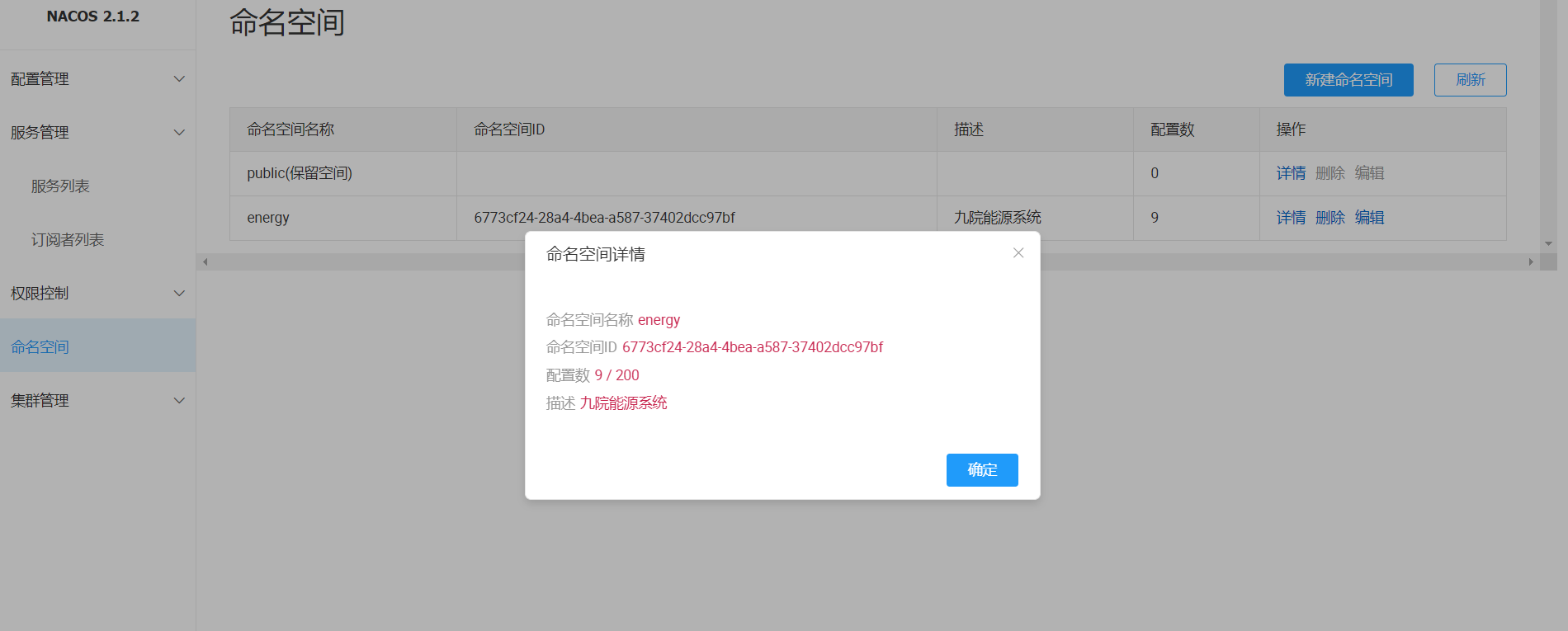




### 初始化配置

#### 创建命名空间





填写对应配置：

命名空间id:

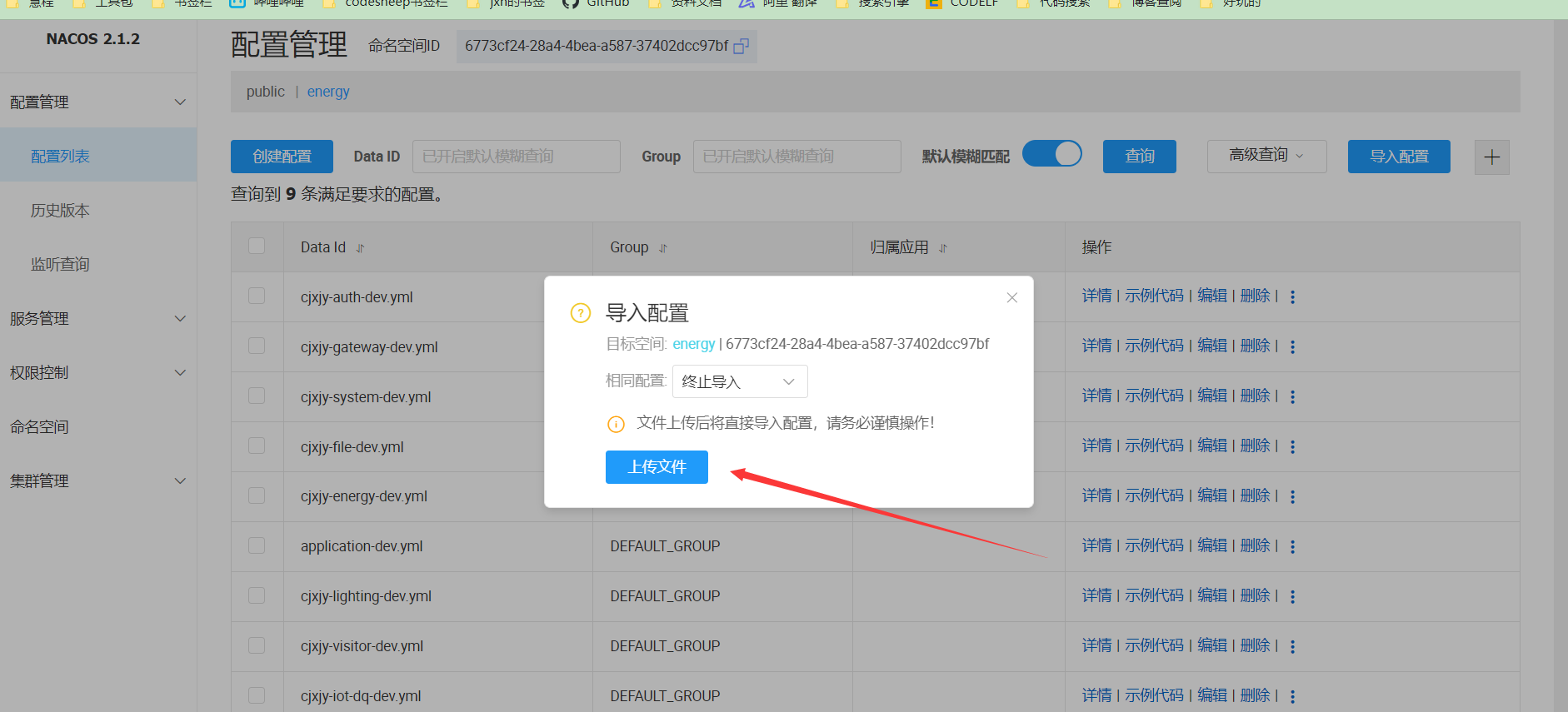
|  |  |
| --- | --- |
| 6773cf24-28a4-4bea-a587-37402dcc97bf |  |

命名空间名： energy

描述： 九院能源系统

### 在配置管理/配置列表中导入配置

将 cjxjy能源服务/nacos-config 中文件导入 配置

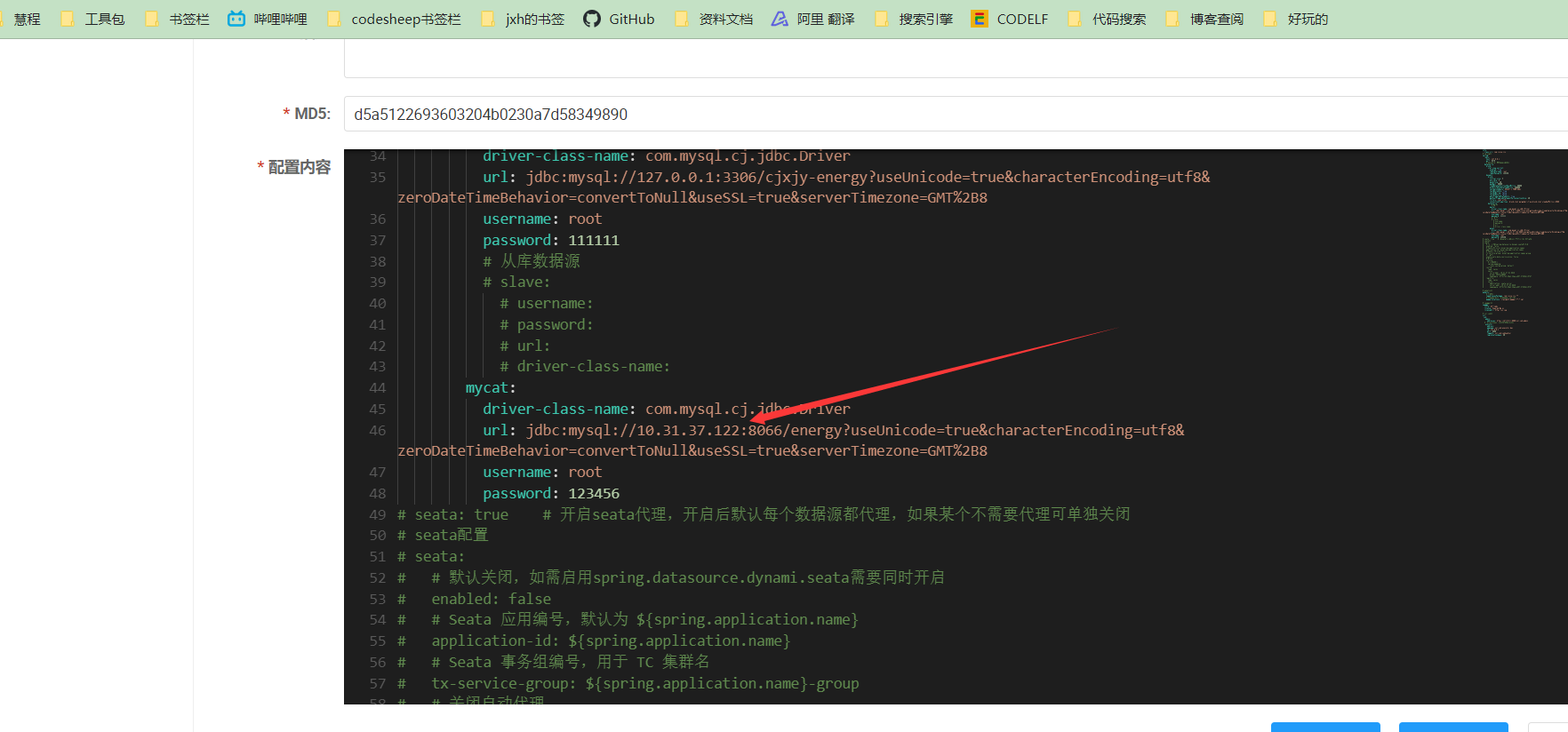


对于其中

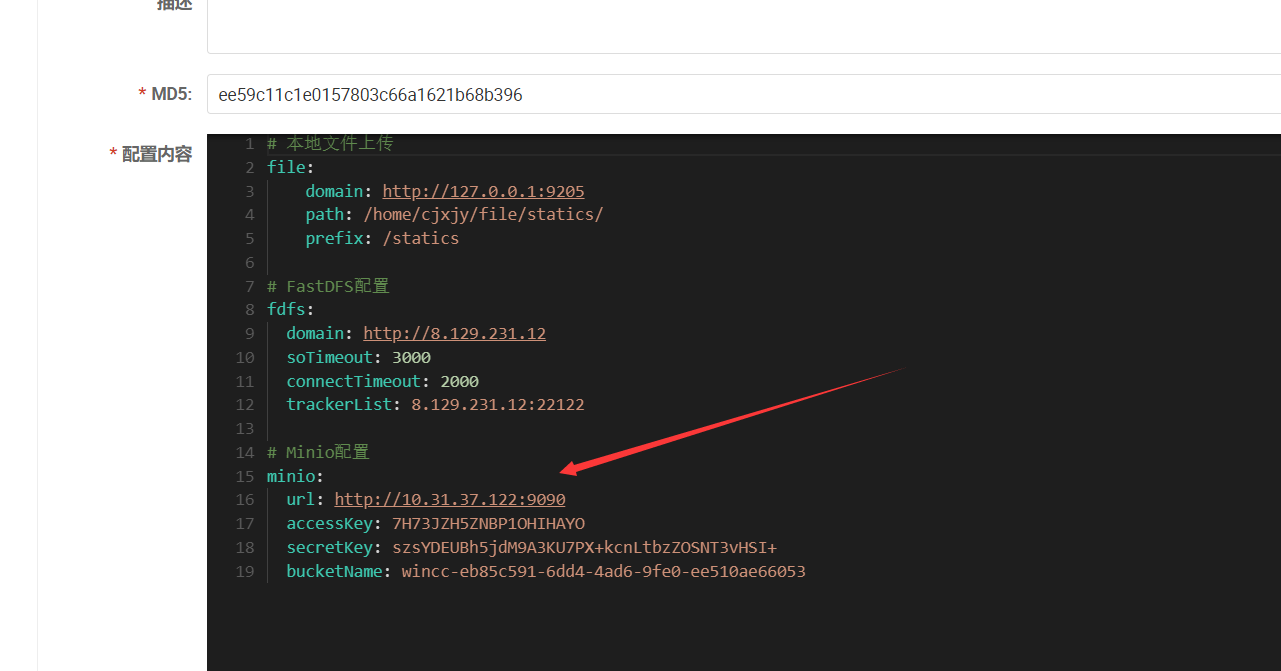
**Cjxjy-energy-dev.yml**

**cjxjy-visitor-dev.yml**

中的mycat 设置，需要修改对应的ip



对于其中的 cjxjy-file-dev.yml 的minio 配置，修改对应ip



## xxl-job-2.3.0

### 解压xxl-job镜像

Docker load -I xxl-job-admin.tar

### 初始化sql脚本

执行脚本： 部署\linux镜像\init-sql-container\xxl\_job-init.sql

### 启动容器

docker run -d -p 8080:8080 --name xxl-job-admin \

-e PARAMS=" \

--spring.datasource.url=jdbc:mysql://10.31.37.122:3306/xxl\_job?Unicode=true&characterEncoding=UTF-8 \

--spring.datasource.username=root \

--spring.datasource.password=cjxjy123456" \

-v /usr/local/xxl-job-2/admin/logs:/data/applogs \

--privileged=true \

xuxueli/xxl-job-admin:2.3.0

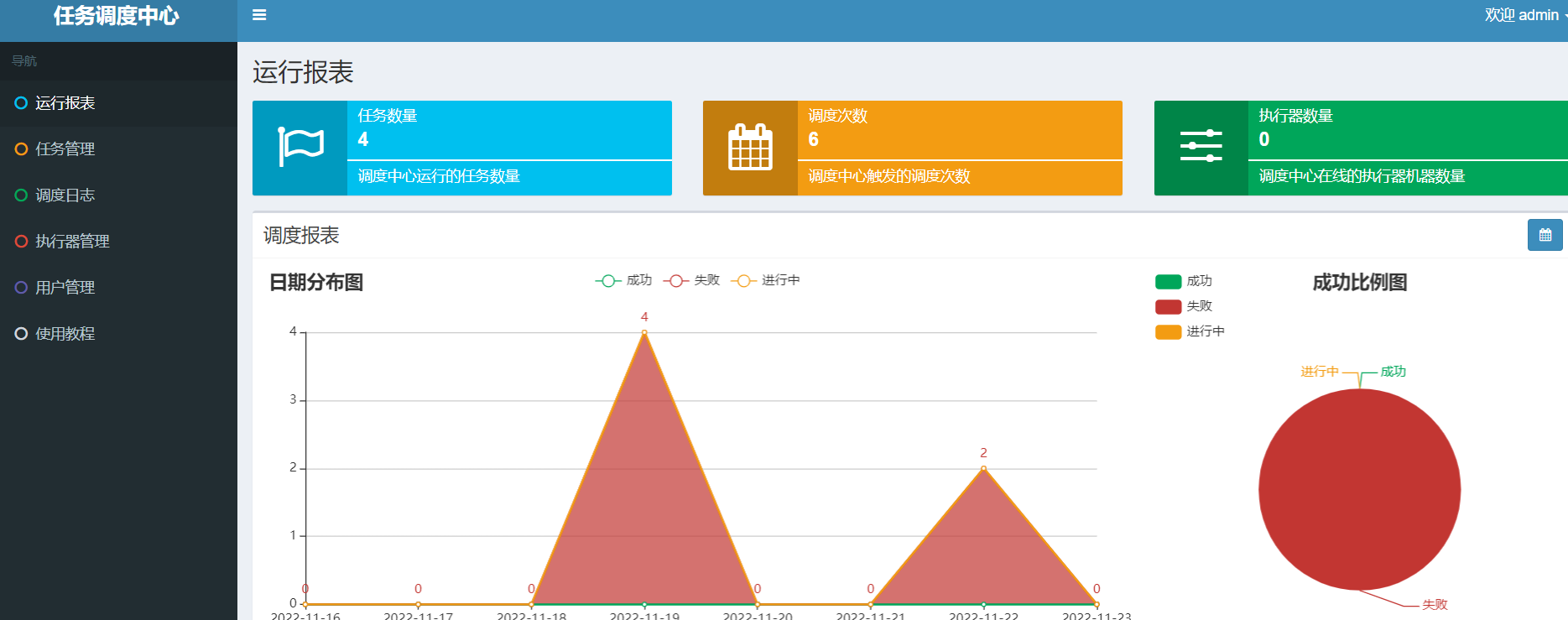
启动请切换自己的数据库ip，docker启动无法使用localhost或127.0.0.1

### 测试访问

http://10.31.37.122:8080/xxl-job-admin/

用户名： admin

密码： 123456



## Minio

### 解压加载minio 镜像

Docker load -I minio.tar

### 启动minio

docker run -d -p 9000:9000 -p 9090:9090 --name minio \ --restart=always \ -e MINIO\_ACCESS\_KEY=minio \ -e MINIO\_SECRET\_KEY=minio@123 \ -v /usr/local/minio/data:/data \ -v /usr/local/minio/config:/root/.minio \ minio/minio server /data --console-address ":9000" --address ":9090"

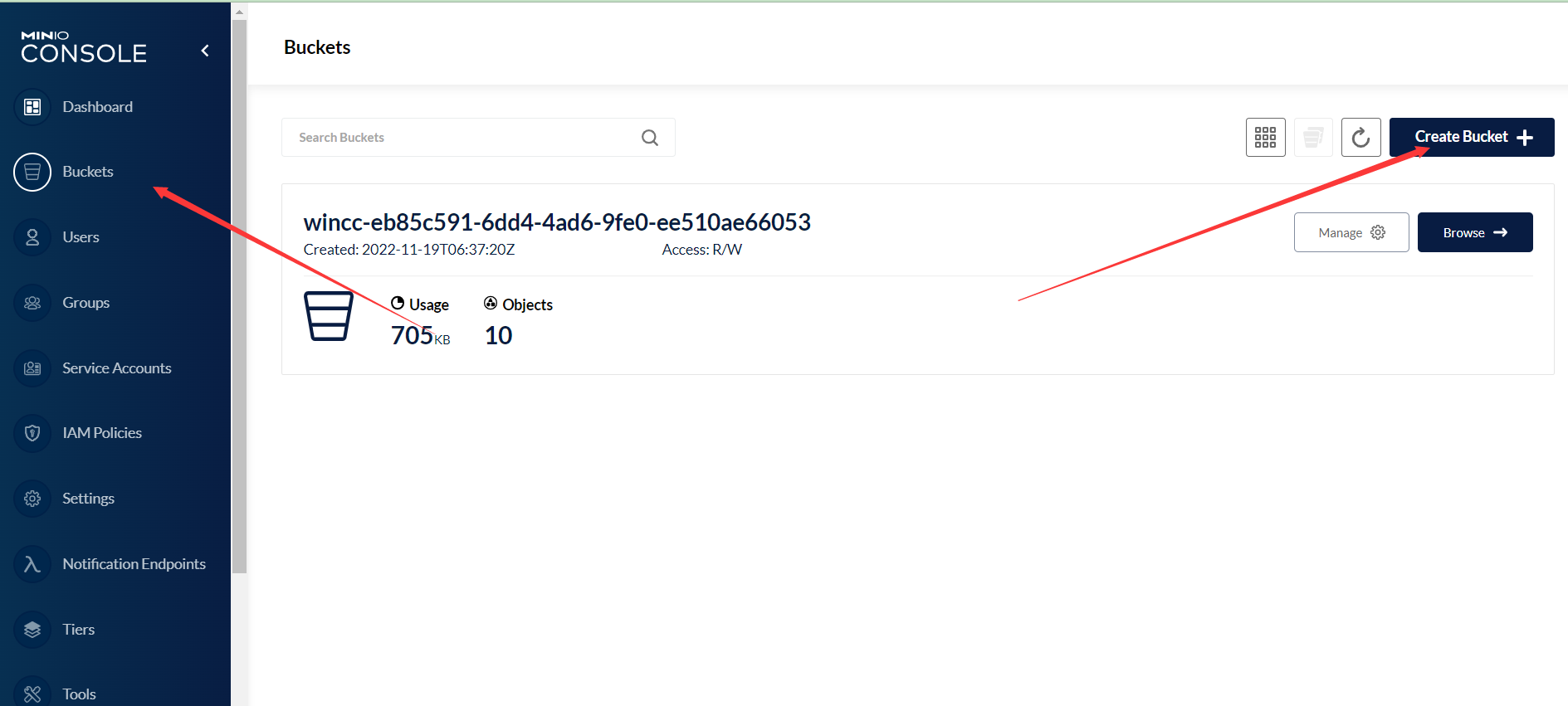
### 测试访问：

<http://10.31.37.122:9000>

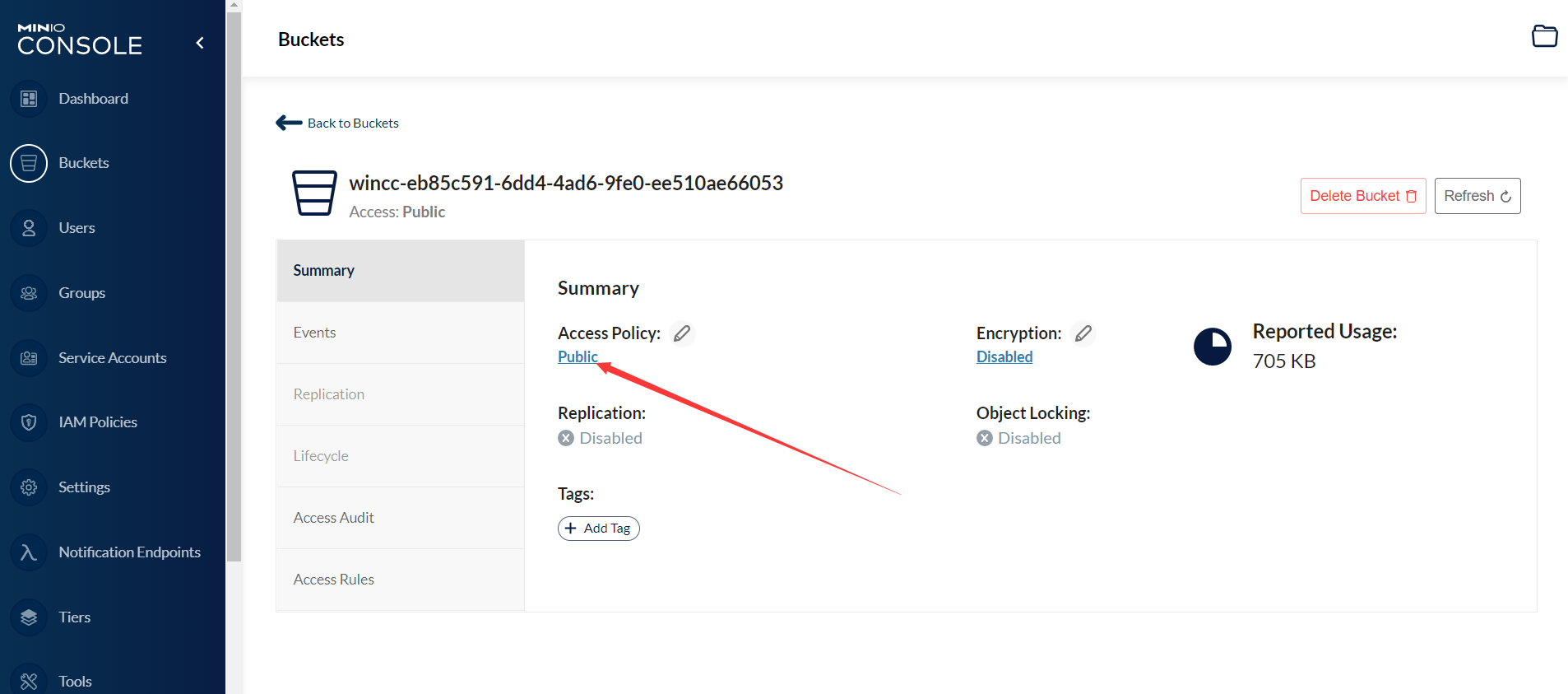
初始用户名密码： minio minio@123

### 初始化数据仓

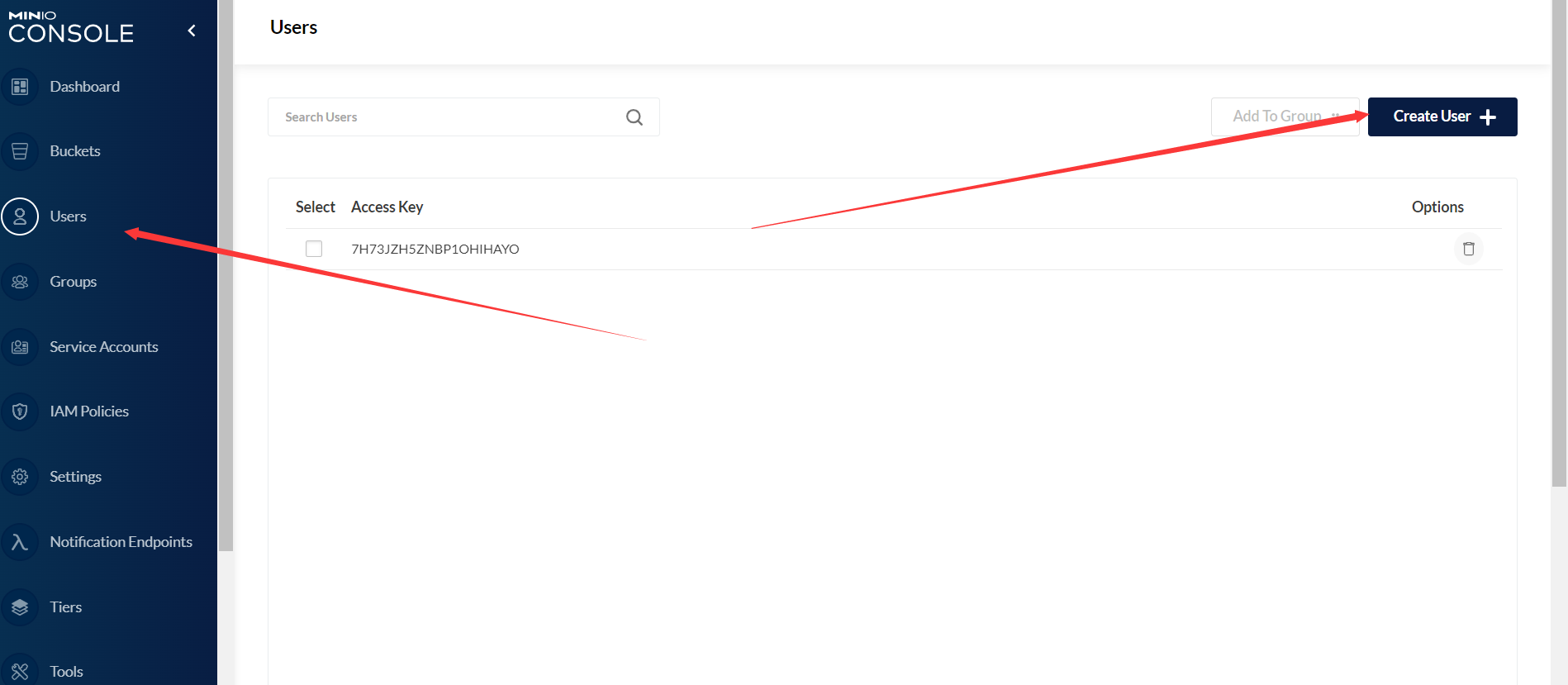
创建bucket

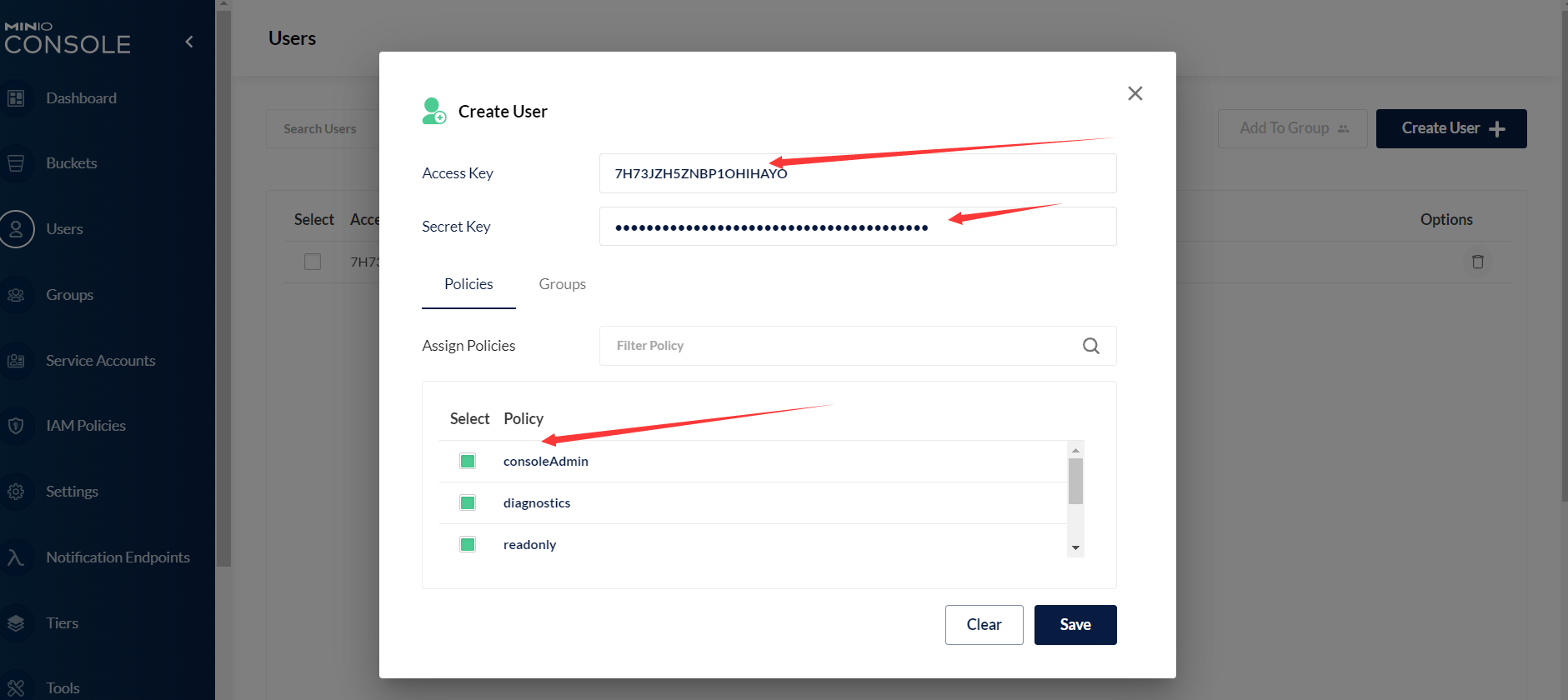


Bucket name: wincc-eb85c591-6dd4-4ad6-9fe0-ee510ae66053

修改访问权限为：public 

### 创建user





填写正确的用户名密码：

accessKey: 7H73JZH5ZNBP1OHIHAYO

secretKey: szsYDEUBh5jdM9A3KU7PX+kcnLtbzZOSNT3vHSI+

并给出所需权限，建议全选

## Rabbitmq

### 解压并加载rabbitmq镜像

docker load -i rabbitmq-stomp.tar

此rabbitmq 默认开启了 stomp 插件。

### 启动rabbitmq服务

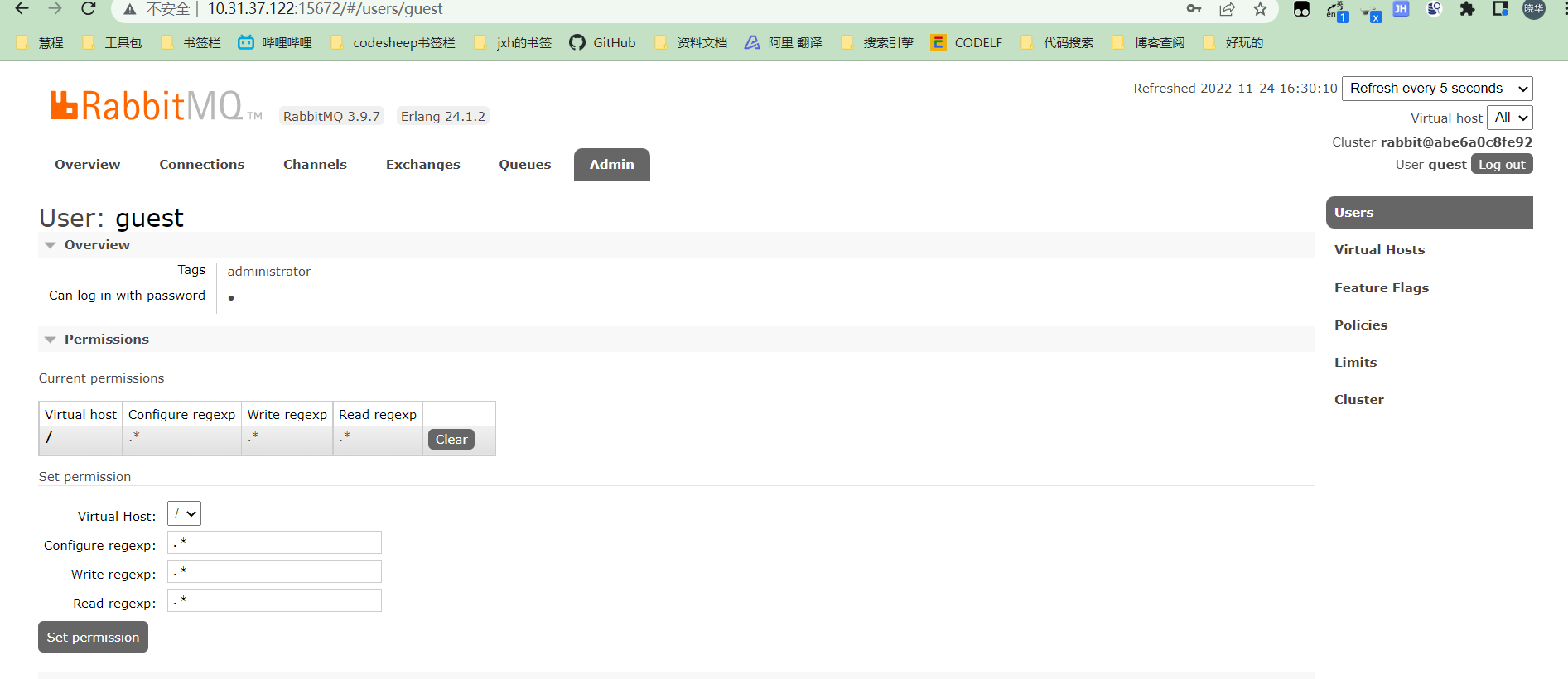
docker run -d -p 15672:15672 -p 5672:5672 -p 15674:15674 \ --name rabbitmq-server \ --restart=always \ --privileged=true \ rabbitmq-stomp:stomp

### 访问测试是否成功

10.31.37.122:15672

初始账号密码：

guest guest



# Java应用

## 初始化数据库

部署\cjxjy能源服务\init-sql

将其中的sql文件执行

然后再执行 update-sql 即可

将对应的部署包安装， 之后进入执行 startup.sh 即可

## cjxjy-gateway

## cjxjy-auth

## cjxjy-system

## cjxjy-file

## cjxjy-energy

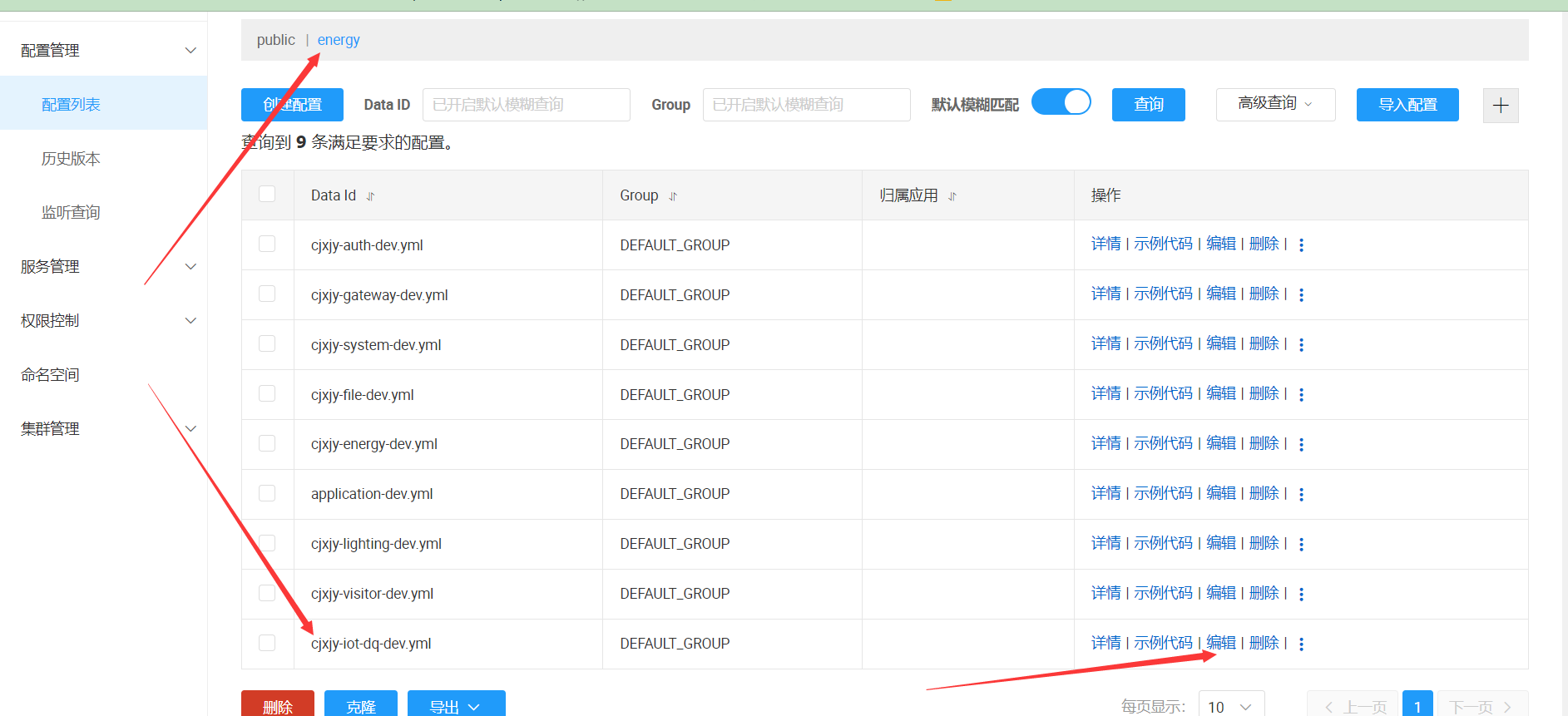
## cjxjy-lighting

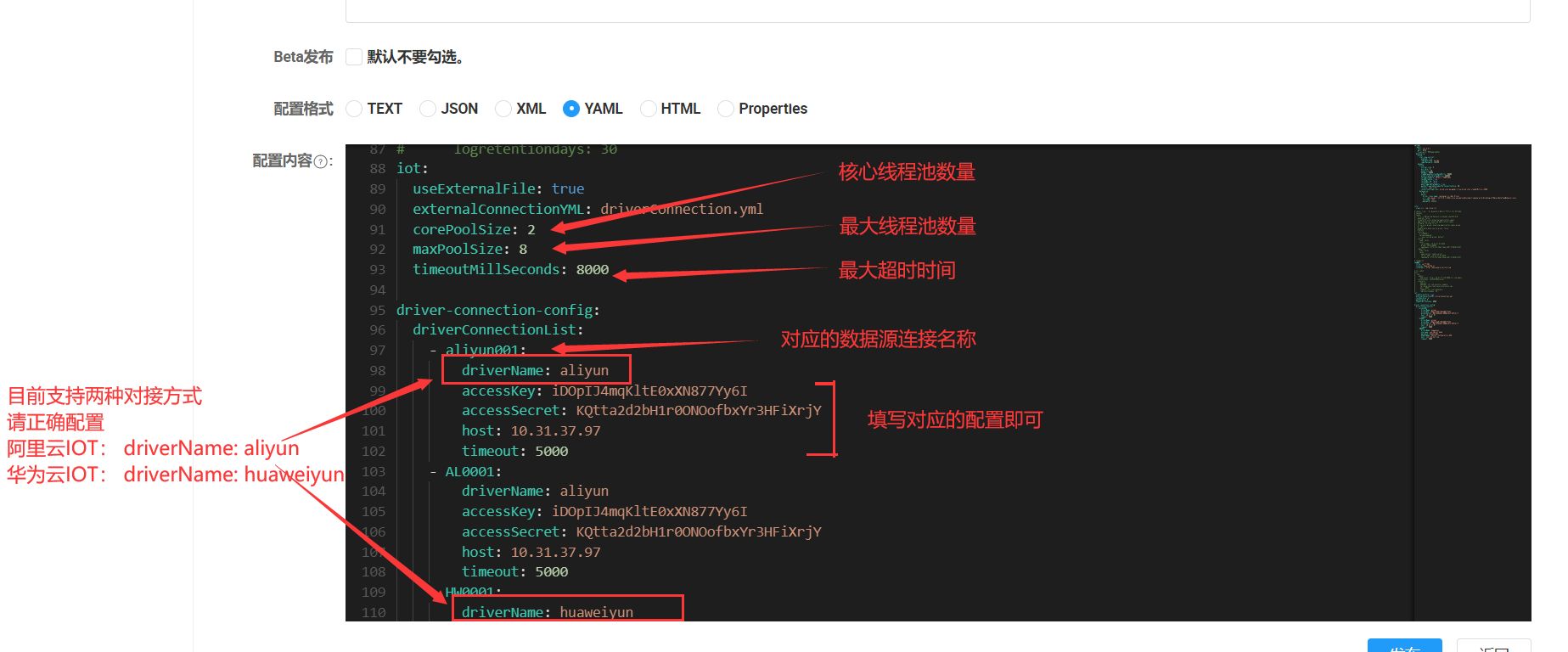
## cjxjy-visitor

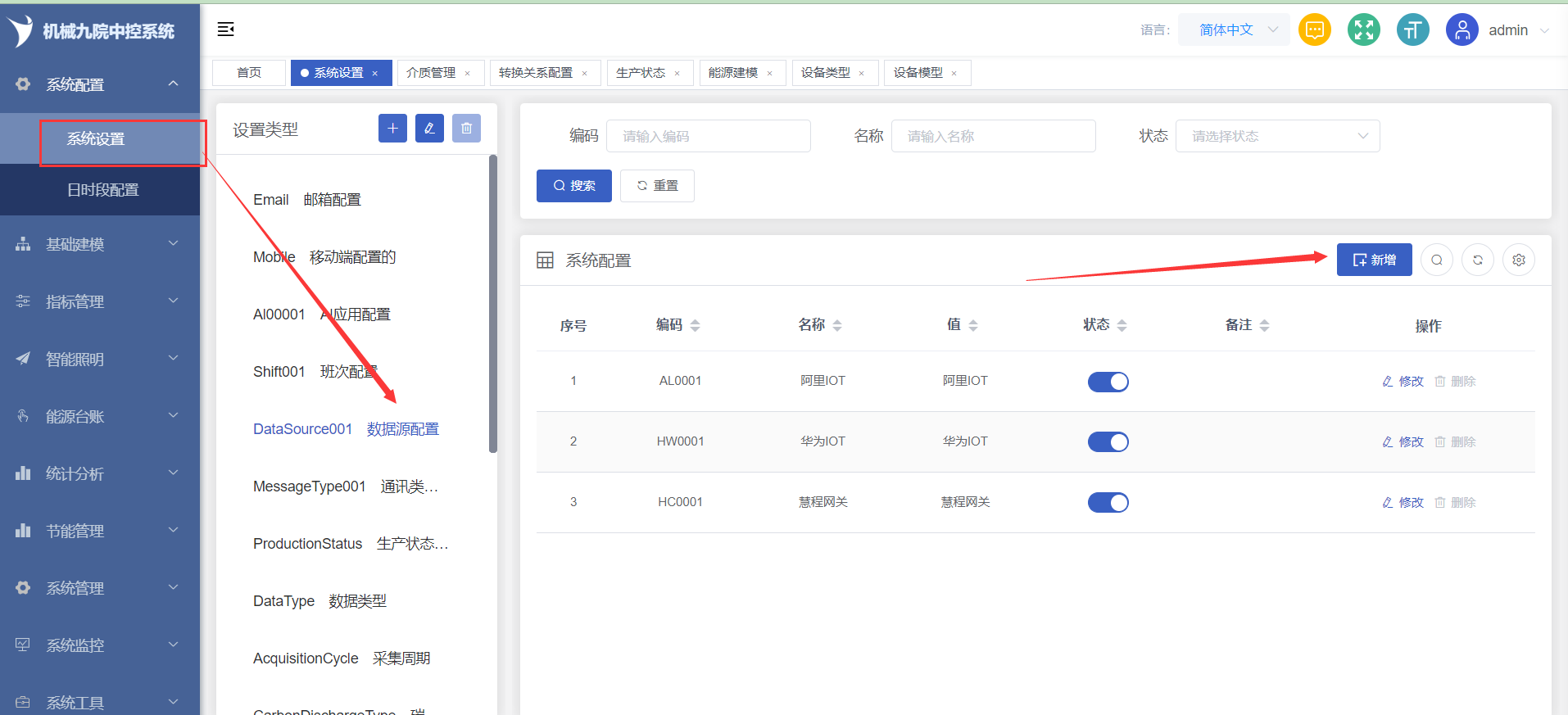
## cjxjy-iot-dq

### iot-dq iot数据查询系统需要调整配置

在部署成功的nacos中，修改对应配置







在平台中添加数据源， 数据源编码即为数据查询服务中的数据源连接名称

# Web资源部署

## 第一次部署：

\部署\linux基础服务\nginx.conf

将nginx 中的 nginx.conf 文件进行覆盖

位置： /usr/local/nginx/conf

执行：**nginx -s reload**