Klustron集群弹性扩容

**注意：**

 如无特别说明，文中的版本号可以使用任何已发布版本的版本号代替。所有已发布版本详见：<http://doc.klustron.com/zh/Release_notes.html>。

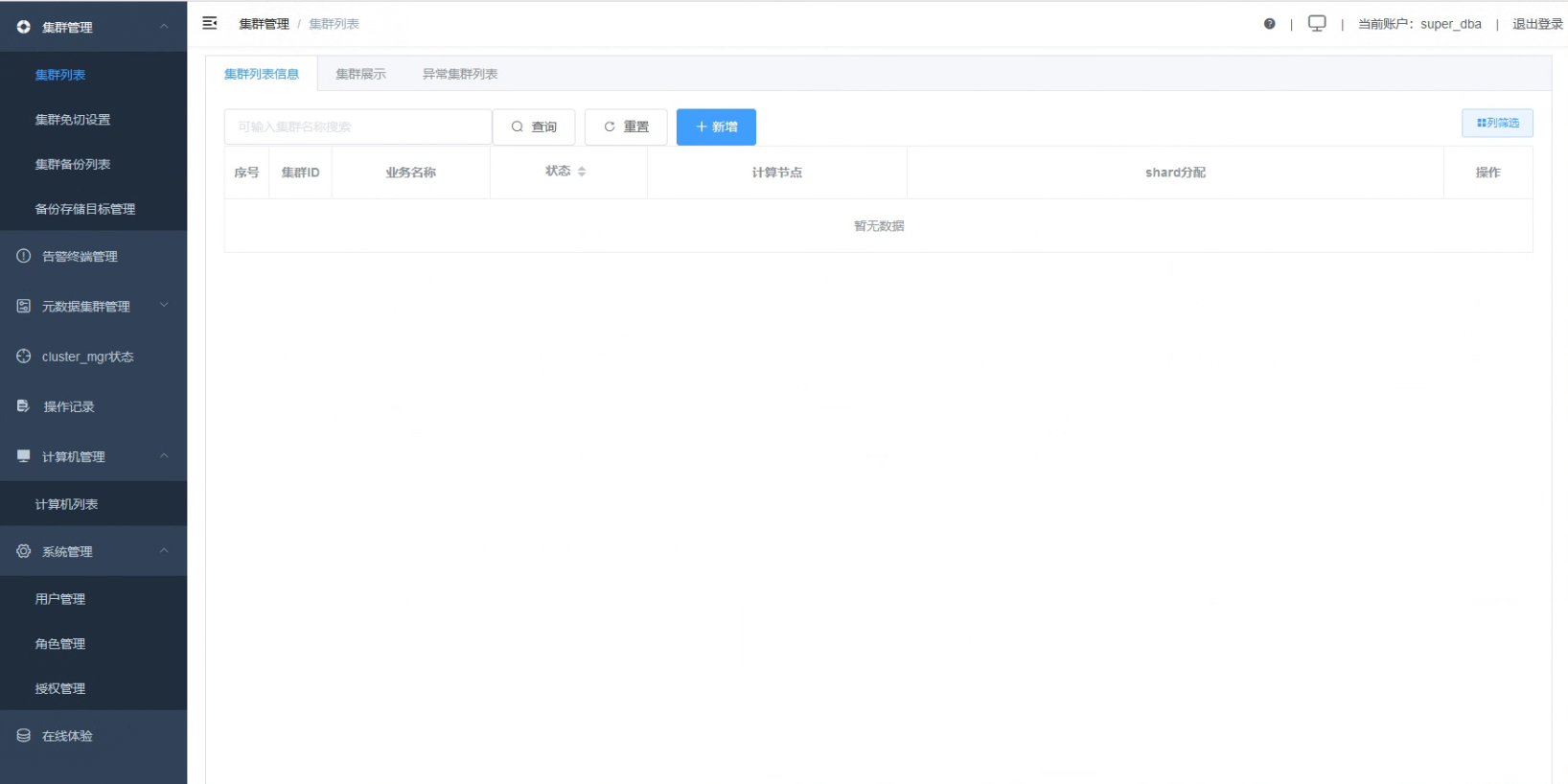
**本文目标：**

主要内容是如何在Klustron数据库中创建集群、在集群数据库创建分区表、集群中增加存储shard节点、将数据库中的分区表部分分区扩容到新增加的shard节点上、最后在数据库中查看扩容后分区表的所有分区分布情况；以及在集群中扩容计算节点。所有的测试都是通过XPanel控制台和PostgreSQL客户端连接集群完成。

XPanel服务安装的服务器IP是192.168.56.112。在能访问192.168.56.112的机器上打开浏览器，输入地址：http://192.168.56.112:18080/KunlunXPanel/#/login?redirect=%2Fdashboard

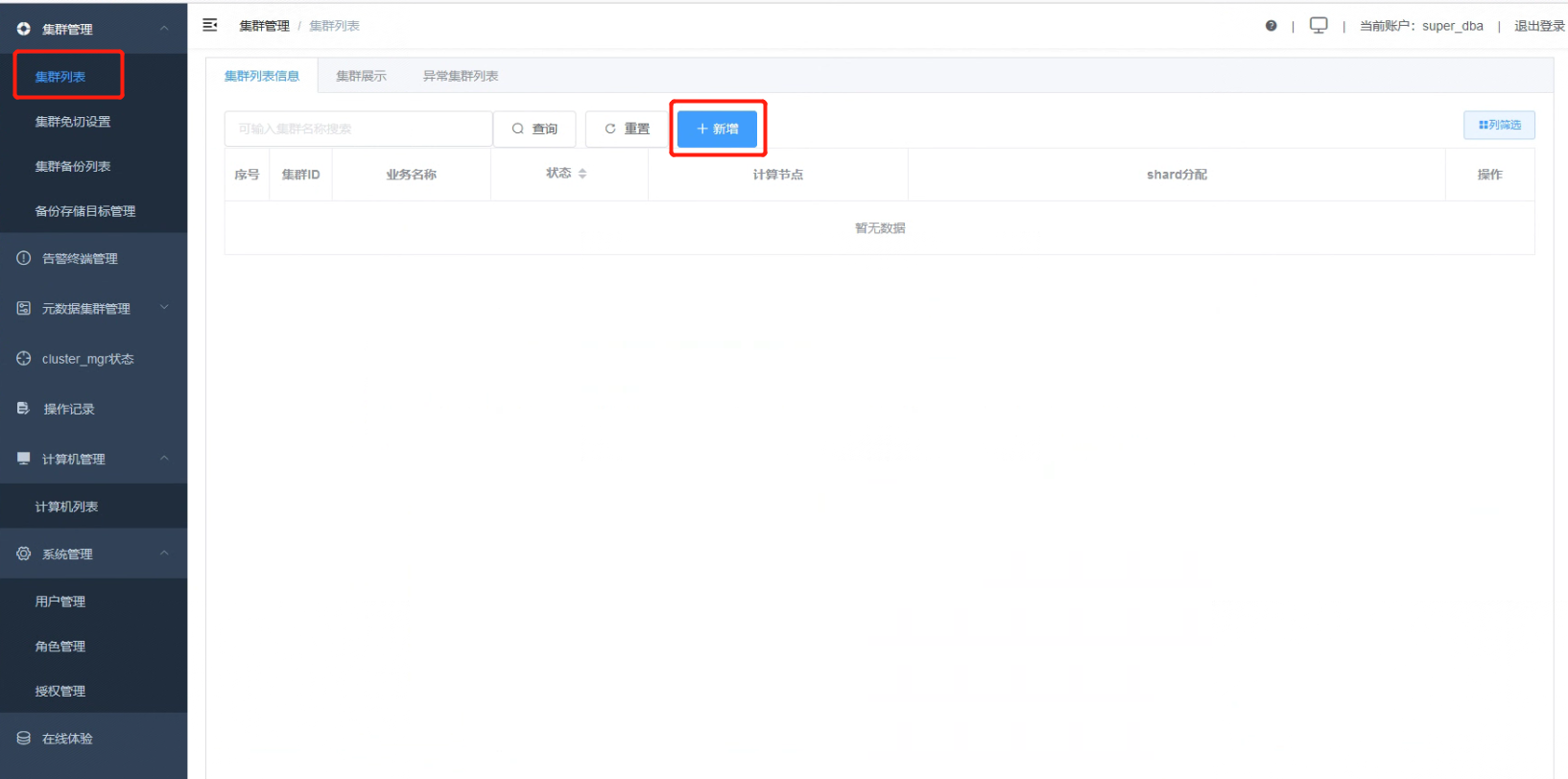
初次登录用户名和密码是：super\_dba/super\_dba，初次登陆需要修改super\_dba密码。

登录后首页显示如下：

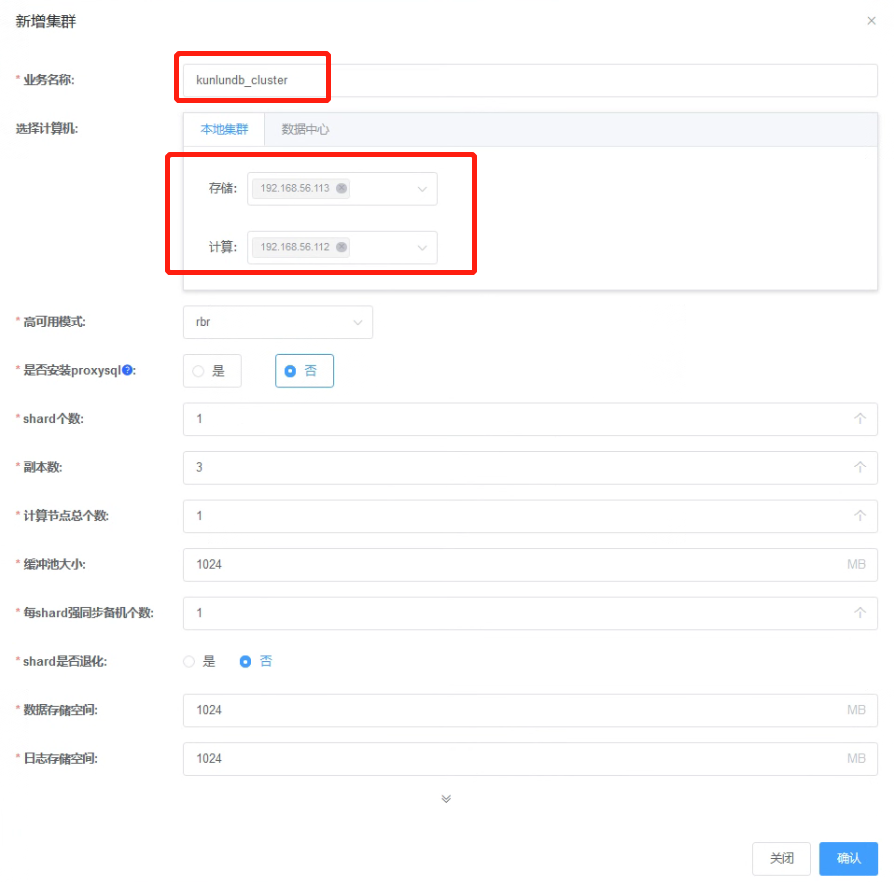


**1 创建集群**

1.1 点击“集群管理”、“集群列表”，集群列表界面点击“新增”按钮。



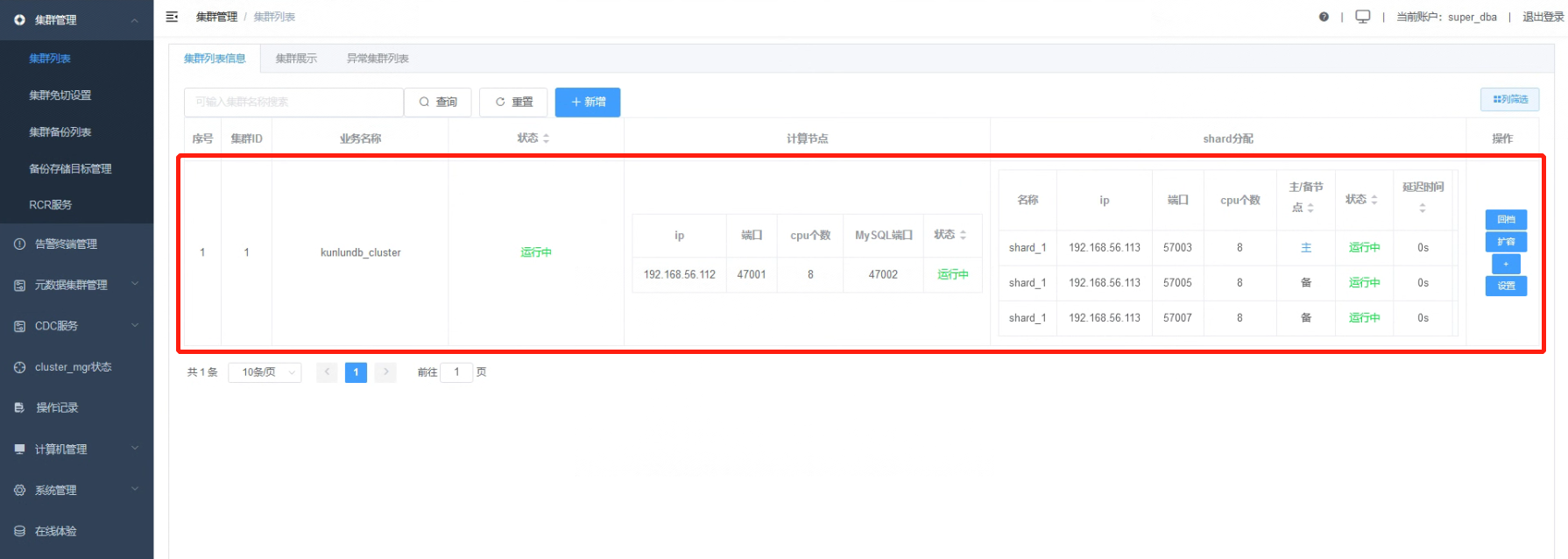
1.2 添加集群参数如下图：



1.3 点击“确认”，查看创建集群任务的完成情况。



1.4 所有状态都是“运行中”， 集群创建完成。



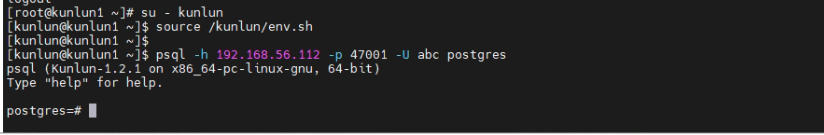
**2 准备试验数据**

2.1 打开一个连向集群kunlundb\_cluster（计算节点IP：192.168.56.112）的ssh终端窗口，以kunlun用户身份登陆该主机，然后用PostgreSQL客户端连接集群数据库。

[root@kunlun1 ~]# su - kunlun

[kunlun@kunlun1 ~]$ source /kunlun/env.sh

[kunlun@kunlun1 ~]$ psql -h 192.168.56.112 -p 47001 -U abc postgres



2.2 在数据库中创建测试表，并插入测试数据。

postgres=# create table prod\_part (id int primary key, name char(8)) partition by hash(id);

postgres=# create table prod\_part\_p1 partition of prod\_part for values with (modulus 6, remainder 0);

postgres=# create table prod\_part\_p2 partition of prod\_part for values with (modulus 6, remainder 1);

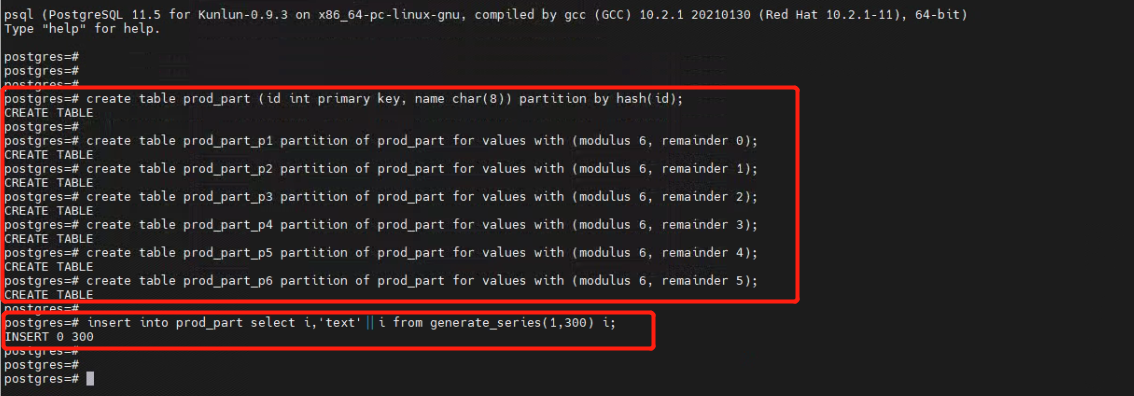
postgres=# create table prod\_part\_p3 partition of prod\_part for values with (modulus 6, remainder 2);

postgres=# create table prod\_part\_p4 partition of prod\_part for values with (modulus 6, remainder 3);

postgres=# create table prod\_part\_p5 partition of prod\_part for values with (modulus 6, remainder 4);

postgres=# create table prod\_part\_p6 partition of prod\_part for values with (modulus 6, remainder 5);

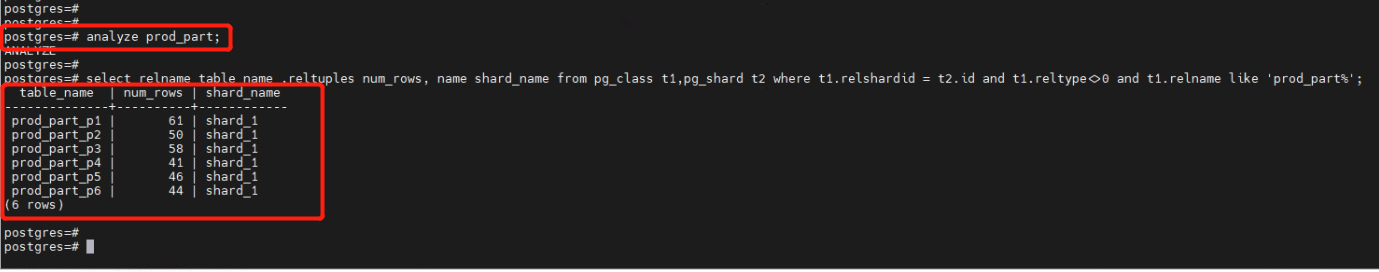
postgres=# insert into prod\_part select i,'text'||i from generate\_series(1,300) i;



2.3 查看数据的分布情况。

postgres=# analyze prod\_part;

postgres=# select relname table\_name ,reltuples num\_rows, name shard\_name from pg\_class t1,pg\_shard t2 where t1.relshardid = t2.id and t1.reltype<>0 and t1.relname like 'prod\_part%';

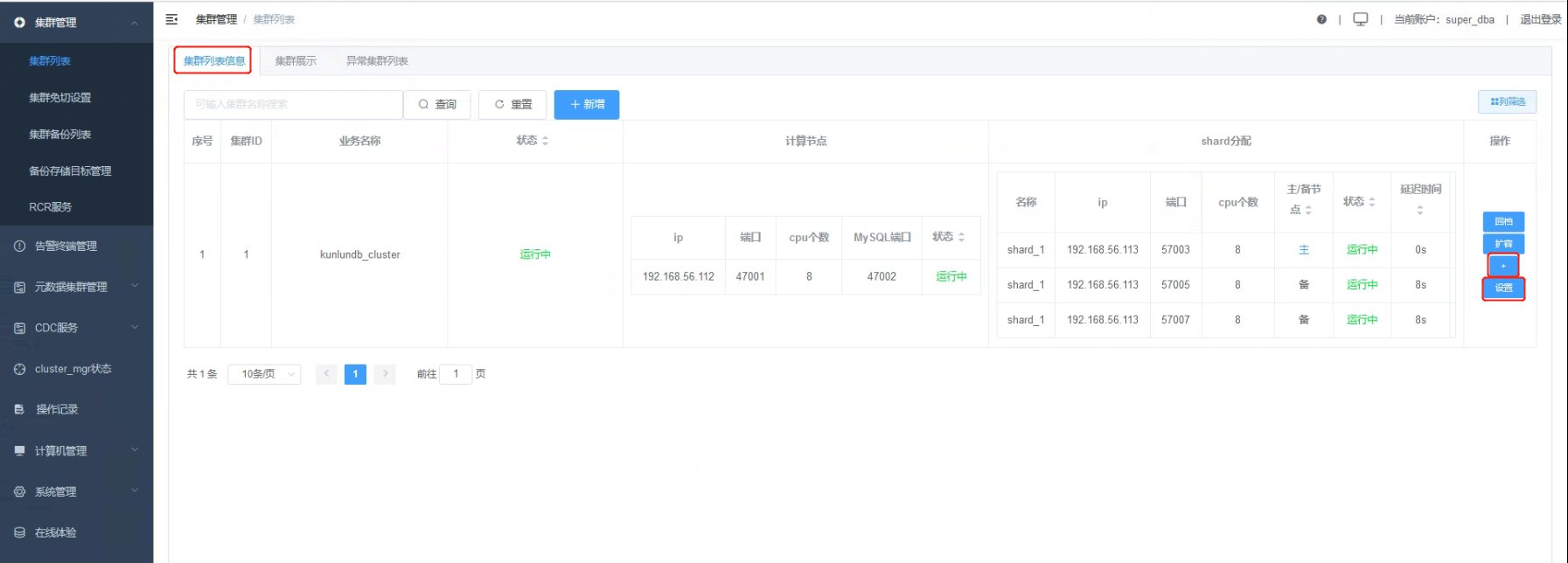


从上面可以看到分区表prod\_part中，prod\_part\_p1、prod\_part\_p2、prod\_part\_p3、prod\_part\_p4、prod\_part\_p5和prod\_part\_p6所有的分区都存放在存储节点shard\_1上。

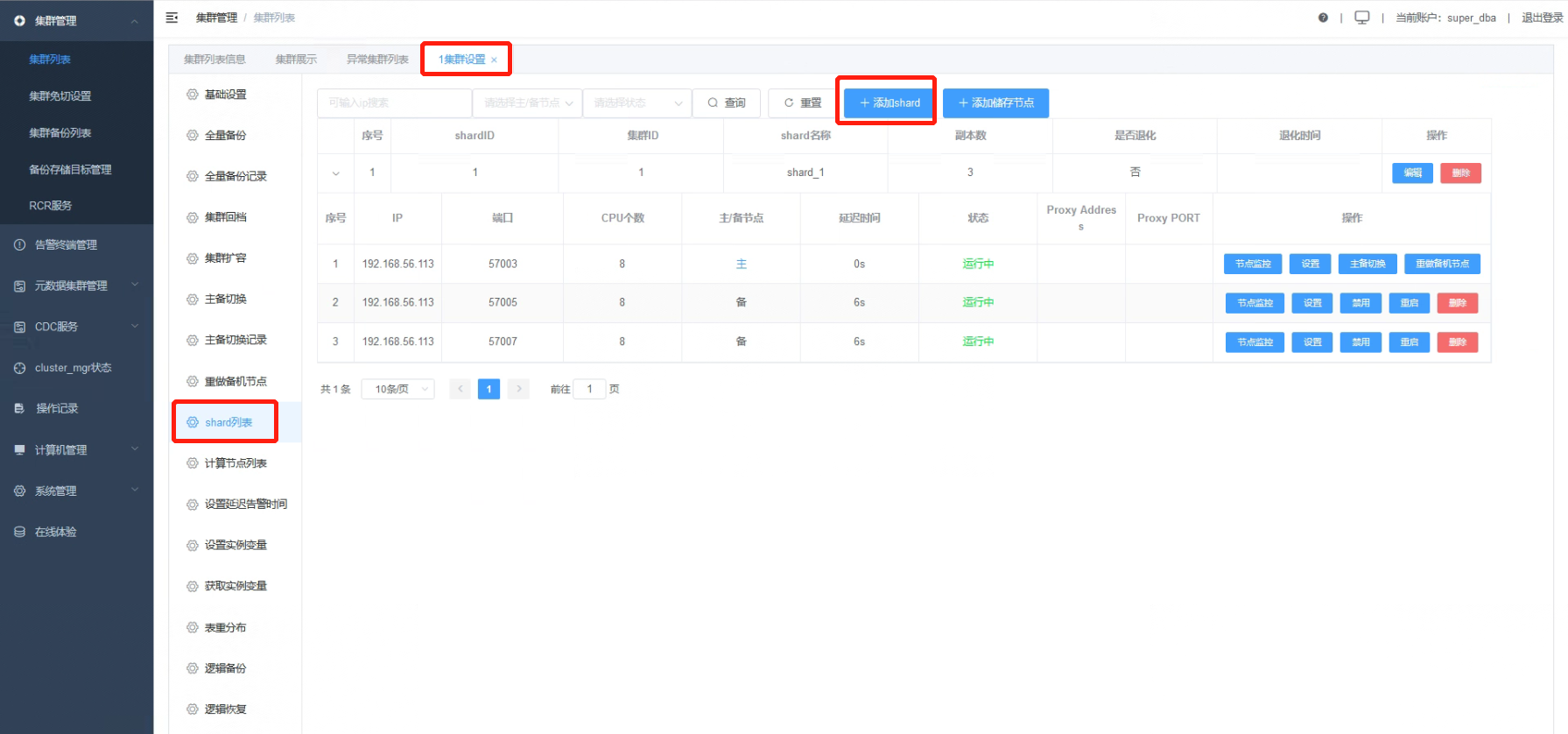
**3 集群扩容存储shard节点**

**3.1 增加存储shard节点**

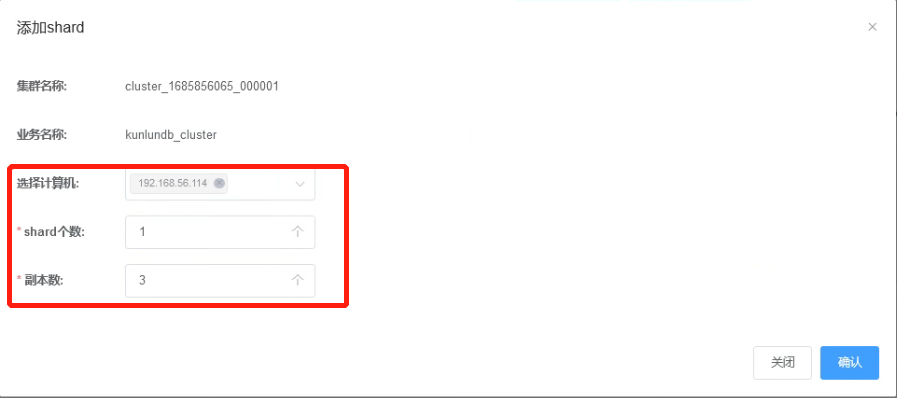
3.1.1 在集群列表信息，对需要增加存储shard节点的集群点击“设置”或“+”按钮。



3.1.2 在“集群设置”中点击“shard列表”，然后点击“添加shard”按钮。



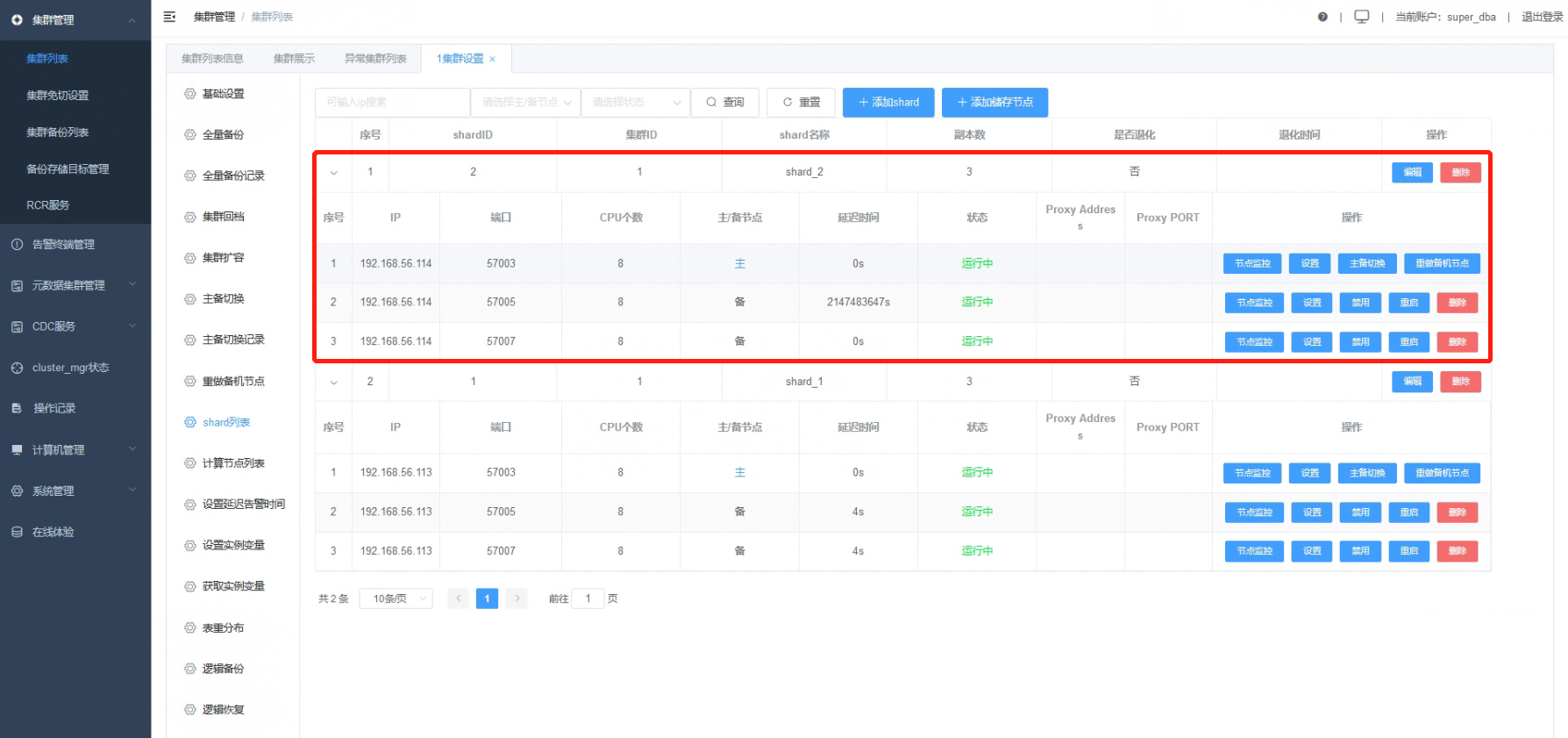
3.1.3 添加shard节点参数如下图。



3.1.4 点击“确认”，查看添加shard节点任务的完成情况。



3.1.5 检查存储shard节点shard\_2已添加到集群。



**3.2 扩容存储shard节点**

3.2.1准备一个python脚本pyprod.py，在集群做存储shard节点扩容的同时对集群数据库持续操作，pyprod.py脚本内容如下：

import psycopg2.extras

from psycopg2 import DatabaseError

import time

import datetime

conn = psycopg2.connect(database='postgres',user='abc',

password='abc',host='192.168.56.112',port='47001')

select\_sql = ''' select \* from prod\_part where id=%s; '''

i = 1

try:

while (i <= 1000) :

cursor = conn.cursor(cursor\_factory=psycopg2.extras.RealDictCursor)

cursor.execute(select\_sql,[i])

res = cursor.fetchall()

print(dict(res[0]))

current\_datetime = datetime.datetime.now()

print("当前日期和时间：", current\_datetime)

if (i == 1000) :

i = 1

else :

i = i+1

cursor.close()

conn.commit()

time.sleep(1)

except (Exception, DatabaseError) as e:

print(e)

input('Press any key and Enter to continue ~!')

conn = psycopg2.connect(database='postgres', user='abc',

password='abc', host='192.168.56.112', port='47001')

select\_sql = ''' select \* from prod\_part where id=%s; '''

while (i <= 1000):

cursor = conn.cursor(cursor\_factory=psycopg2.extras.RealDictCursor)

cursor.execute(select\_sql, [i])

res = cursor.fetchall()

print(dict(res[0]))

current\_datetime = datetime.datetime.now()

print("当前日期和时间：", current\_datetime)

if (i == 1000):

i = 1

else:

i = i + 1

cursor.close()

conn.commit()

time.sleep(1)

finally:

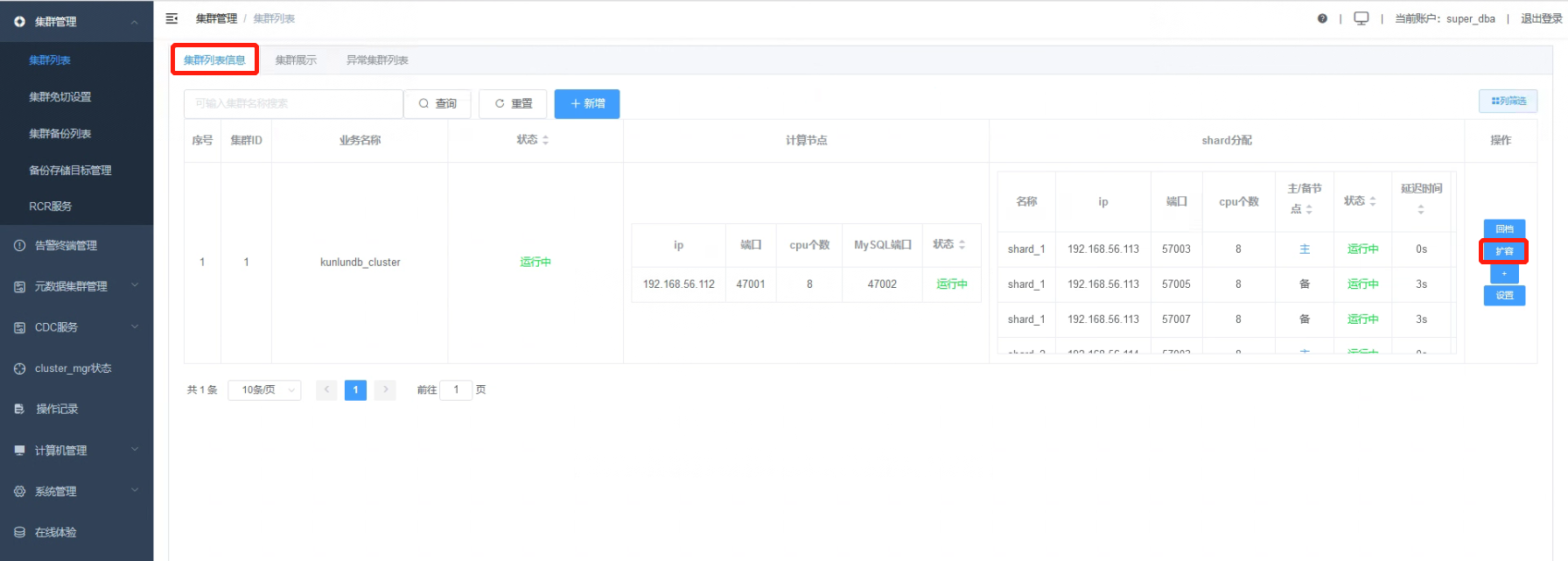
conn.close()

3.2.2运行pyprod.py脚本，持续对数据库进行操作

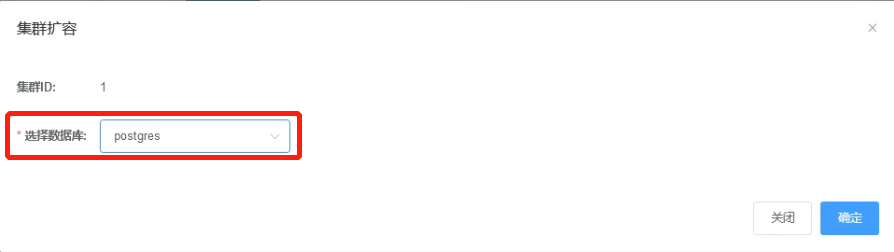
[kunlun@kunlun1 scripts]$ python pyprod.py



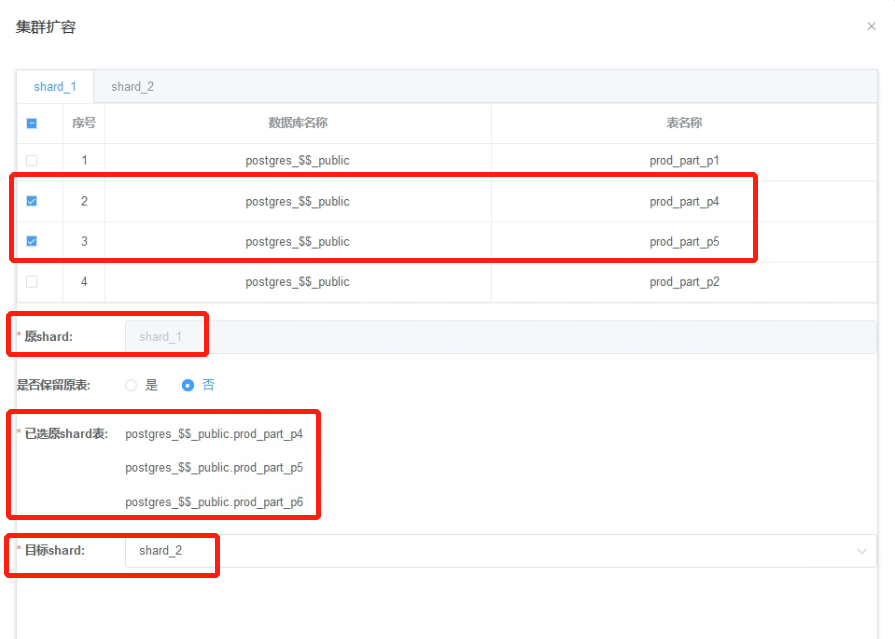
3.2.3 在集群列表信息栏，对需要做shard之间表分区转移的集群，点击“扩容”按钮。



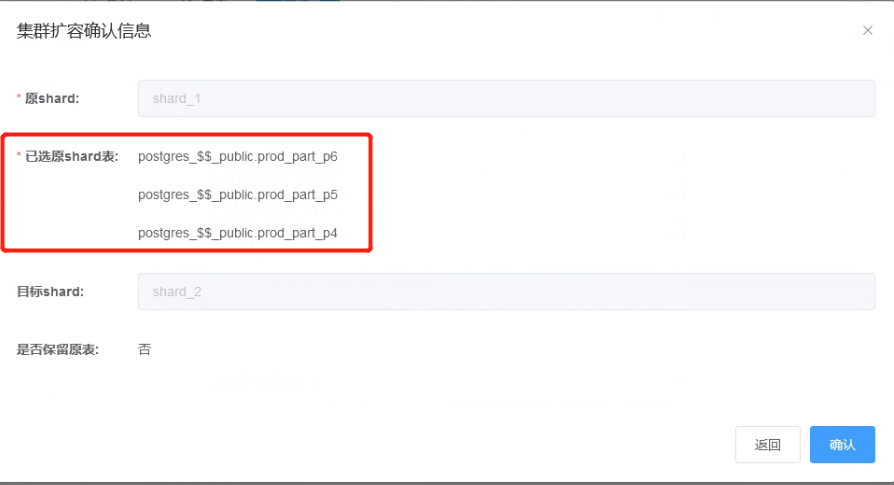
3.2.4 在集群扩容“选择数据库”的下拉列表选择需要扩容的数据库，然后点击 “确定”按钮。例如本示例“postgres”数据库。



3.2.5 选择要从shard\_1要移动到shard\_2上的表，这里选择了prod\_part\_p4、prod\_part\_p5、prod\_part\_p6三张分区表；在是否保留原表选择“否”，目标shard选择“shard\_2”，然后点击“提交”。



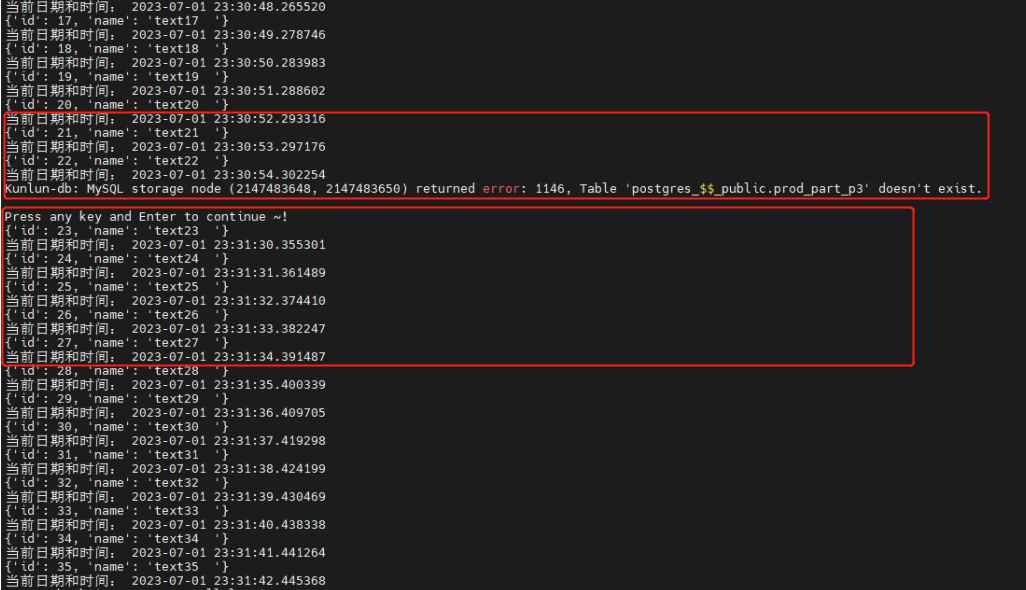
3.2.6 在集群扩容确认信息界面点击“确认”按钮，开始进行集群扩容。



3.2.7 扩容成功。



3.2.7在扩容的过程中，当移动的表是应用程序正在操作的表，会报被操作的表不存在，会有一个短暂的时间不可访问，当移动完成后，应用程序可以继续访问操作这些表。



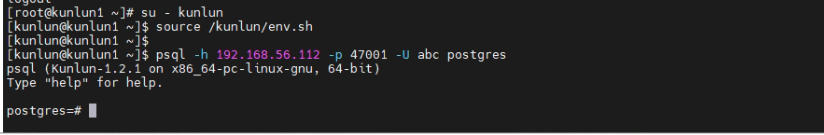
**3.3 扩容后数据校验**

3.3.1 打开一个连向集群kunlundb\_cluster（计算节点IP：192.168.56.112）的ssh终端窗口，以kunlun用户身份登陆该主机，然后用PostgreSQL客户端连接集群，执行数据查询指令。

[root@kunlun1 ~]# su - kunlun

[kunlun@kunlun1 ~]$ source /kunlun/env.sh

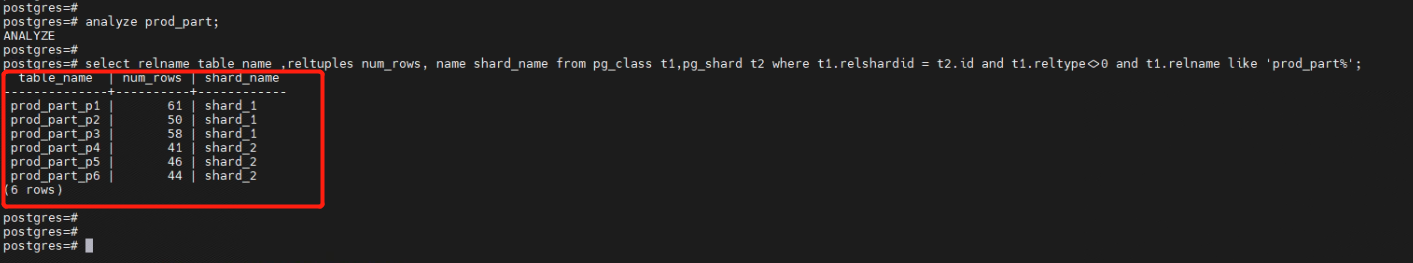
[kunlun@kunlun1 ~]$ psql -h 192.168.56.112 -p 47001 -U abc postgres



3.3.2 查看数据的分布情况。

postgres=# analyze prod\_part;

postgres=# select relname table\_name ,reltuples num\_rows, name shard\_name from pg\_class t1,pg\_shard t2 where t1.relshardid = t2.id and t1.reltype<>0 and t1.relname like 'prod\_part%';

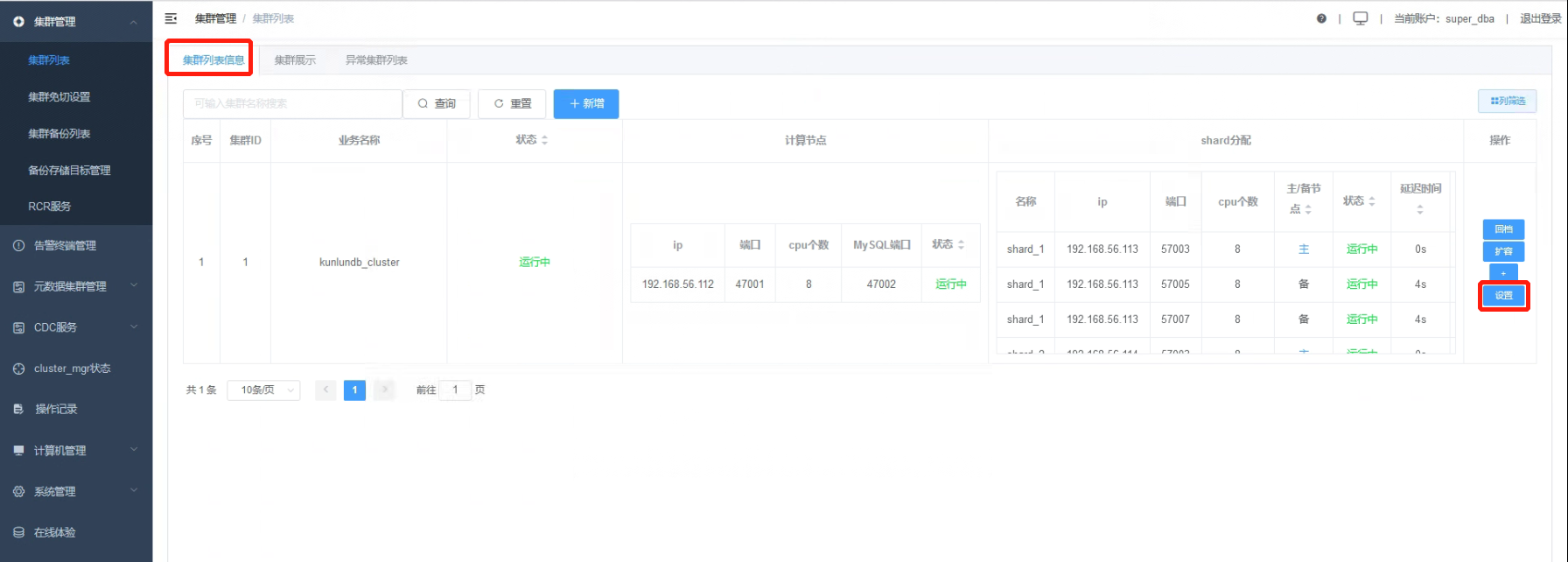


从上面可以看到分区表prod\_part中，prod\_part\_p1、prod\_part\_p2和prod\_part\_p3分区存放在存储节点shard\_1上，而prod\_part\_p4、prod\_part\_p5和prod\_part\_p6分区已扩容存放到存储节点shard\_2上。

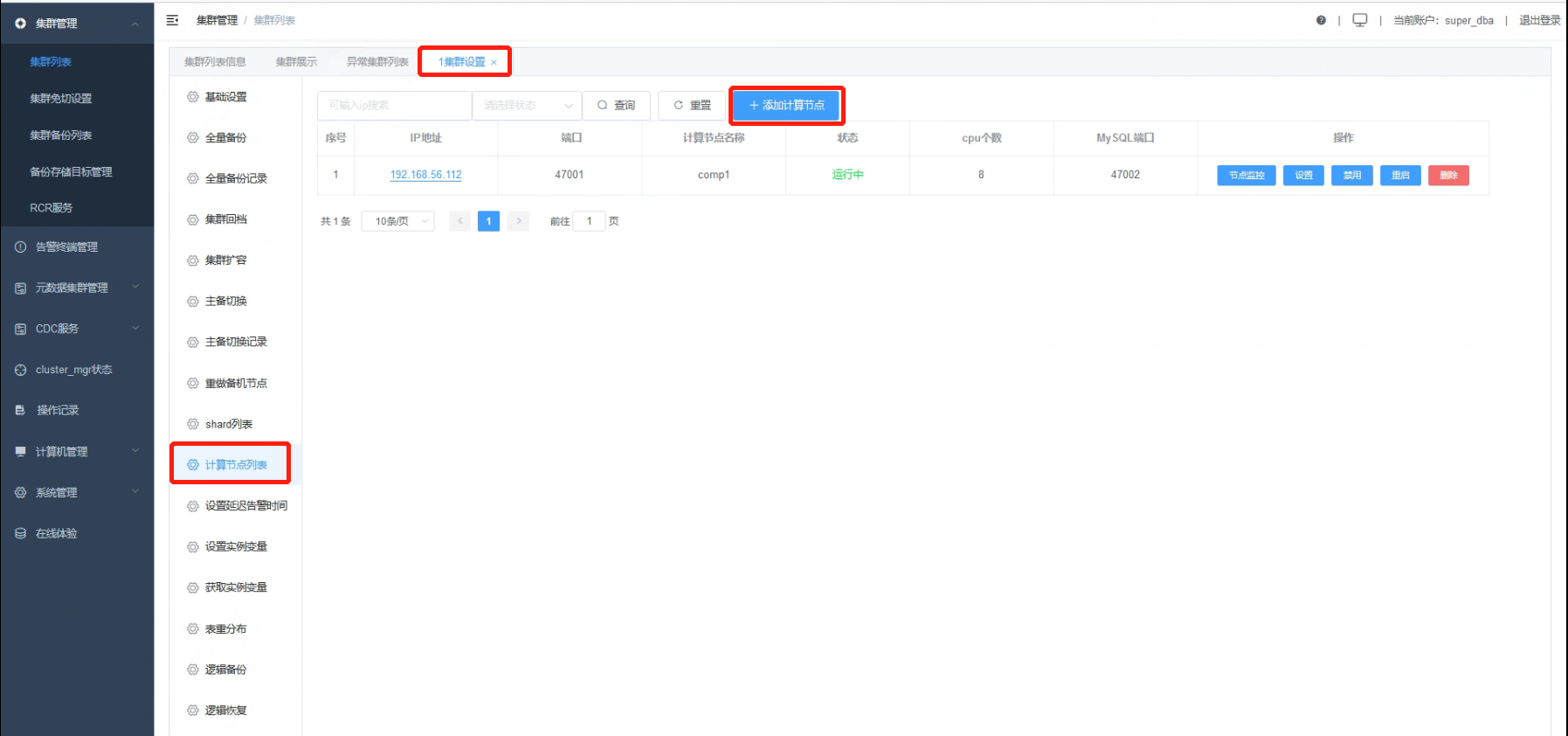
完成集群存储shard节点的扩容。

**4 集群扩容计算节点**

4.1在集群列表信息，对需要增加计算节点的集群点击“设置”按钮。



4.2 在“集群设置”中点击“计算节点列表”，然后点击“添加计算节点”按钮。



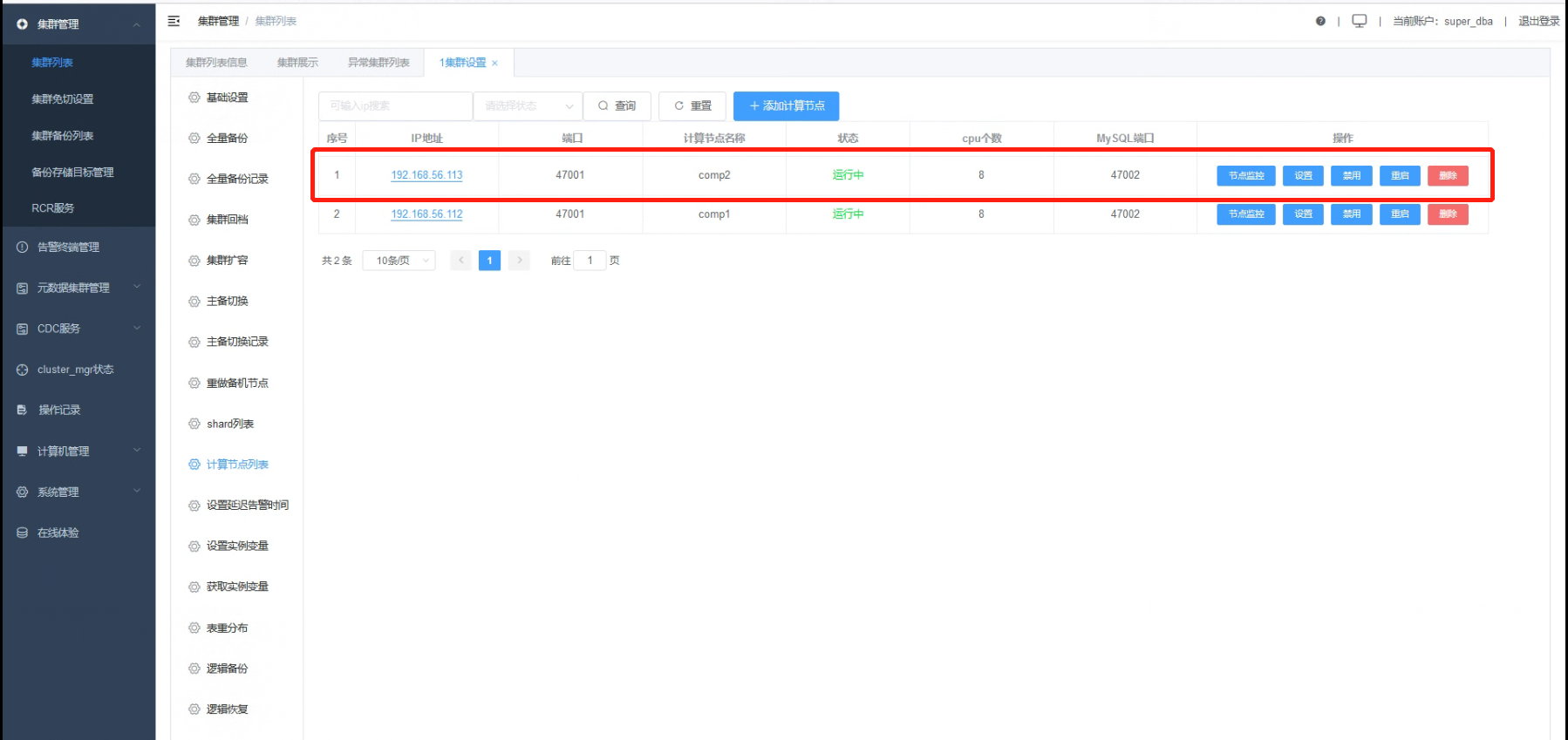
4.3 添加计算节点参数如下图。



4.4 点击“确认”，查看添加计算节点任务的完成情况。



4.5 检查计算节点列表，新的计算节点已添加到集群。



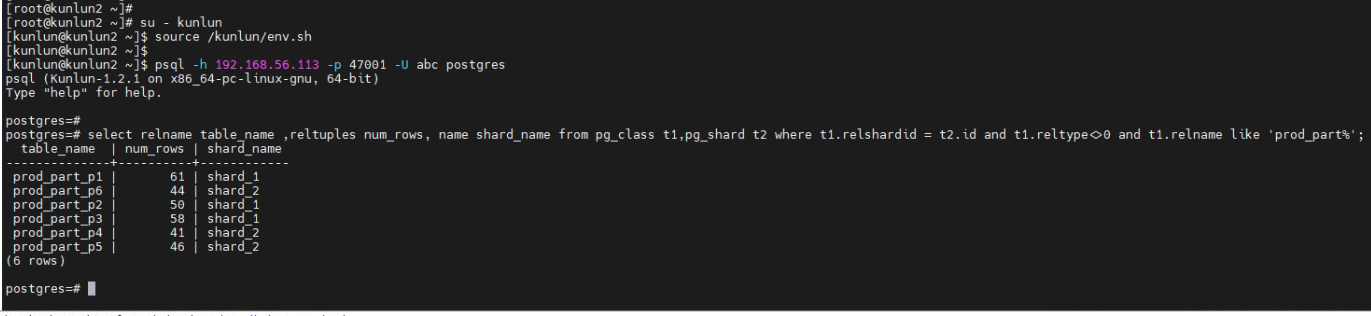
4.6打开一个连向集群kunlundb\_cluster（计算节点IP：192.168.56.113）的ssh终端窗口，以kunlun用户身份登陆该主机，然后用PostgreSQL客户端连接集群数据库（通过新添加的计算节点（192.168.56.113）访问数据库）。

[root@kunlun2 ~]# su - kunlun

[kunlun@kunlun2 ~]$ source /kunlun/env.sh

[kunlun@kunlun2 ~]$ psql -h 192.168.56.113 -p 47001 -U abc postgres

postgres=# select relname table\_name ,reltuples num\_rows, name shard\_name from pg\_class t1,pg\_shard t2 where t1.relshardid = t2.id and t1.reltype<>0 and t1.relname like 'prod\_part%';



完成集群计算节点的扩容。

[END]