

postgresql-集群管理器-pgclusteradmin

作者: [PostgreSQL 中国社区](http://www.postgres.cn)---阿弟

<http://www.postgres.cn>

QQ:4893310

Email:4893310@qq.com

Pgclusteradmin 是一款基于 go 开发的 postgresql 集群管理工具,当前主要功能有“节点资料集中管理”、“参数在线配置,参数文件多版本管理,参数文件模板管理”、“服务管理(即服务 start,stop,restart,stop)”、“vip 管理”、“备机唤醒”、“主备切换”、“巡检报告”;系统支持多用户,操作认证,支持 SSH 密码登陆和 SSH 公钥、私钥登陆;操作人员通过浏览器从远程登录进入管理平台,前面的界面使用 easyui 实现。

一、功能列表

- 节点资料增加,编辑,删除。
- 各个节点运行参数配置,参数文件多版本管理,参数文件模板管理。
- 单一节点服务 start、stop、restart、reload 及显示服务状态。
- 节点对应机器 VIP 绑定与解绑。
- 备机唤醒管理。
- 主备节点一键切换。
- 巡检报告生成及导出
- 管理工具——进程管理
- 管理工具——表锁管理
- 管理工具——查询统计

二、部署环境

Ip: 192.168.0.231
os: centos 6.9
golang: go version go1.8.3 linux/amd64
Postgresql: 9.6.1

三、开发环境部署配置

1、源码安装 golang

```
[root@ad ~]# wget https://storage.googleapis.com/golang/go1.8.3.linux-amd64.tar.gz
[root@ad ~]# tar xzf go1.8.3.linux-amd64.tar.gz -C /usr/local
```

2、配置 golang 环境变量

```
[root@ad go]# vim /etc/profile
```

```
GOROOT=/usr/local/go
GOBIN=$GOROOT/bin
```

```
PATH=$PATH:$GOBIN
```

```
GOPATH=/home/go
```

```
export GOROOT GOBIN GOPATH PATH
```

保存退出

```
[root@ad go]# source /etc/profile
```

```
[root@nfs go]# go version
```

```
go version go1.8.3 linux/amd64
```

3、安装 postgresql

使用 postgresql 主要是用于存储管理节点资料，操作员资料及操作日志。

——下载源码

```
wget https://ftp.postgresql.org/pub/source/v9.6.1/postgresql-9.6.1.tar.gz
```

——解压

```
tar xzf postgresql-9.6.1.tar.gz
```

——编译

```
cd postgresql-9.6.1
```

```
./configure --prefix=/usr/local/pgsql9.6.1 --with-perl --with-tcl --with-python --with-openssl --with-pam  
--without-ldap --with-libxml --with-libxslt
```

```
gmake
```

```
gmake install
```

——初始化

```
su postgres
```

```
/usr/local/pgsql9.6.1/bin/initdb -D /home/postgres/data9.6.1 -E utf8 -U postgres -W
```

——postgresql.conf 配置

```
listen_addresses = '*'
```

```
log_destination = 'csvlog'
```

```
logging_collector = on
```

——pg_hba.conf 配置

```
# IPv4 local connections:
```

```
host      all             all             192.168.1.0/24      md5
```

配置完成后需要重启服务,其它参数视需要自己配。

4、建立 pgclusteradmin 库并导入建立资料表

```
/usr/local/pgsql9.6.1/bin/psql -h 192.168.1.10 -U postgres -d postgres -p 5432
postgres=# create database pgcluster ENCODING 'utf8' template template0;
\c pgcluster
```

——导入源码包中的 pgcluster.sql 脚本

```
\i pgcluster.sql
```

5、下载 pgclusteradmin 所需要的 go 支持包

——ssh 支持包

```
[root@ad ~]# cd /usr/local/go/src
[root@ad src]# mkdir golang.org
[root@ad src]# cd golang.org/
[root@ad golang.org]# mkdir x
[root@ad src]# cd x/
[root@ad x]# git clone https://github.com/golang/crypto.git
正克隆到 'crypto'...
remote: Counting objects: 3256, done.
remote: Total 3256 (delta 0), reused 0 (delta 0), pack-reused 3255
接收对象中: 100% (3256/3256), 2.31 MiB | 958.00 KiB/s, done.
处理 delta 中: 100% (2106/2106), done.
```

——session 支持包

```
[root@ad x]# cd /usr/local/go/src
[root@ad src]# mkdir github.com
[root@ad src]# cd github.com
[root@ad github.com]# mkdir astaxie
[root@ad github.com]# cd astaxie/
[root@ad astaxie]# git clone https://github.com/astaxie/session
正克隆到 'session'...
remote: Counting objects: 50, done.
remote: Total 50 (delta 0), reused 0 (delta 0), pack-reused 50
Unpacking objects: 100% (50/50), done.
[root@ad astaxie]# ll
总用量 8
```

——postgresql 操作支持包

```
[root@ad pgclusteradmin]# cd /usr/local/go/src/github.com/
[root@ad github.com]# mkdir pkg
[root@ad github.com]# cd pkg
[root@ad pkg]# git clone https://github.com/pkg/errors
Initialized empty Git repository in /usr/local/go1.8.3/src/github.com/pkg/errors/.git/
remote: Counting objects: 418, done.
remote: Total 418 (delta 0), reused 0 (delta 0), pack-reused 418
Receiving objects: 100% (418/418), 86.42 KiB, done.
Resolving deltas: 100% (253/253), done.
[root@ad pkg]#
```

```
[root@ad astaxie]# cd /usr/local/go/src/github.com/
[root@ad github.com]# mkdir jackc
[root@ad github.com]# cd jackc
[root@ad jackc]# git clone https://github.com/jackc/pgx
正克隆到 'pgx'...
remote: Counting objects: 3613, done.
remote: Compressing objects: 100% (243/243), done.
remote: Total 3613 (delta 157), reused 0 (delta 0), pack-reused 3370
接收对象中: 100% (3613/3613), 1.24 MiB | 228.00 KiB/s, done.
处理 delta 中: 100% (2481/2481), done.
```

——golang excel 驱动包

```
[root@ad github.com]# cd /usr/local/go/src/github.com/
[root@ad github.com]# mkdir tealeg
[root@ad github.com]# cd tealeg/
[root@ad tealeg]# git clone https://github.com/tealeg/xlsx
正克隆到 'xlsx'...
remote: Counting objects: 2793, done.
remote: Total 2793 (delta 0), reused 0 (delta 0), pack-reused 2792
接收对象中: 100% (2793/2793), 1.11 MiB | 357.00 KiB/s, done.
处理 delta 中: 100% (1503/1503), done.
```

四、pgclusteradmin 部署

1、下载 pgclusteradmin 源码

```
[root@ad pgclusteradmin]# cd /home/ad
[root@ad ad]# git clone https://github.com/chenaisheng/pgclusteradmin
正克隆到 'pgclusteradmin'...
remote: Counting objects: 374, done.
remote: Compressing objects: 100% (177/177), done.
remote: Total 374 (delta 201), reused 348 (delta 185), pack-reused 0
接收对象中: 100% (374/374), 284.09 KiB | 197.00 KiB/s, done.
```

处理 delta 中: 100% (201/201), done.

[root@ad ad]#

2、配置连接数据库参数

打开 pgclusteradmin.go 文件，拉下最后面，找到函数 extractConfig()，代码如下所示

/*

功能描述：配置 postgresql 连接参数

参数说明：无

返回值说明：

pgx.ConnConfig -- pg 连接参数结构体

*/

```
func extractConfig() pgx.ConnConfig {

    var config pgx.ConnConfig

    config.Host = "127.0.0.1" //数据库主机 host 或 ip
    config.User = "postgres" //连接用户
    config.Password = "pgsql" //用户密码
    config.Database = "pgcluster" //连接数据库名
    config.Port = 5432 //端口号

    return config
}
```

修改成上面部署 postgresql 的相应参数即可。

3、配置 ssh 登录使用的私钥

怎样配置使用 SSH 公钥、私钥登陆

(1)、使用 root 用户登陆后，运行以下第一句指令，其他根据提示进行输入：

ssh-keygen -t rsa

也可以使用 DSA

Generating public/private rsa key pair.

Enter file in which to save the key (/root/.ssh/id_rsa):

建议直接回车使用默认路径

Created directory '/root/.ssh'

Enter passphrase (empty for no passphrase):

输入密码短语（留空则直接回车）

Enter same passphrase again:

重复密码短语

Your identification has been saved in /root/.ssh/id_rsa.

Your public key has been saved in /root/.ssh/id_rsa.pub.

The key fingerprint is:

05:71:53:92:96:ba:53:20:55:15:7e:5d:59:85:32:e4 root@test

The key's randomart image is:

+--[RSA 2048]-----+

```
|  o o ..          |
| . o oo.+ .       |
| o.+... =         |
| ...o             |
| o S              |
| .                |
|                  |
|                  |
|                  |
+-----+
```

此时在/root/.ssh/目录下生成了 2 个文件，id_rsa 为私钥，id_rsa.pub 为公钥。

将公钥复制到需要登录电脑的/root/.ssh/authorized_keys 文件中。

好了，至此只要你保存好你的私钥，你的服务器相比原来使用 root 用户加密码登陆来说已经安全多了。

(2)、数据库服务管理用户 postgres 的公钥、私钥生成方法与上面 root 用户的公钥、私钥生成方法一致，也可以与 root 共用一对公钥、私钥。

(3)、打开 pgclusteradmin.go 文件，拉下最后面，找到函数 get_postgres_private_key() 和 get_root_private_key()，把数据库服务管理用户和管理员 root 使用的私钥复制上去

4、运行 pgclusteradmin

```
[root@ad ad]# cd pgclusteradmin/
```

```
[root@ad pgclusteradmin]# go run pgclusteradmin.g
```

5、访问 pgclusteradmin

打开一个浏览器，输入 <http://192.168.0.231:10001> 即可进入管理器，192.168.0.231 换成你自己 ip 地址即可。初始化的用户名和密码都是“admin”

6、注册 pgclusteradmin 做为一个服务（线上部署使用，测试的话可以漂过）

1)、编译 pgclusteradmin.go 程序为可执行程序

```
[root@ad pgclusteradmin]# go build pgclusteradmin.go
```

2)、复制 pgclusteradmin 程序及 easyui 目录至/usr/local/pgclusteradmin 目录下

```
[root@ad pgclusteradmin]# mkdir /usr/local/pgclusteradmin
```

```
[root@ad pgclusteradmin]# cp pgclusteradmin easyui/ /usr/local/pgclusteradmin -rf
```

3)、复制服务管理脚本至目录至/etc/init.d 目录下

```
[root@ad pgclusteradmin]# cp ./etc/init.d/pgclusteradmin /etc/init.d/pgclusteradmin
[root@ad pgclusteradmin]# chmod +x /etc/init.d/pgclusteradmin
```

4)、配置 pgclusteradmin 为服务

```
[root@ad pgclusteradmin]# chkconfig --add pgclusteradmin
[root@ad pgclusteradmin]# chkconfig --list | grep pgclusteradmin
pgclusteradmin 0:关闭 1:关闭 2:启用 3:启用 4:启用 5:启用 6:关闭
```

5)、测试 pgclusteradmin 服务使用

```
[root@ad pgclusteradmin]# service pgclusteradmin start
Starting pgclusteradmin
[root@ad pgclusteradmin]# service pgclusteradmin status
pgclusteradmin (pid 15686) already running..
[root@ad pgclusteradmin]# service pgclusteradmin restart
Stopping pgclusteradmin
Starting pgclusteradmin
[root@nfs pgclusteradmin]# service pgclusteradmin status
pgclusteradmin (pid 15717) already running..
[root@ad pgclusteradmin]# service pgclusteradmin help
Usage: /etc/init.d/pgclusteradmin {start|stop|status|restart|help}
        start - start pgclusteradmin
        stop  - stop pgclusteradmin
        status - show current status of pgclusteradmin
        restart - restart pgclusteradmin if running by sending a SIGHUP or start if not running
        help  - this screen
[root@ad pgclusteradmin]#
```

五、使用文档

1、主界面说明

Postgresql集群管理器							
<div> 增加节点 修改资料 删除节点 参数配置 服务管理 VIP绑定 备机唤醒 主备切换 修改登录密码 退出 </div>							
	节点名称	建立时间	节点类别	运行状态	版本信息	主机名或IP	SSH
1	users_cluster_10_9620	2017-03-11 12:24:40.794889	备节点	运行中	PostgreSQL 9.6.1 on	192.168.1.10	22
2	users_cluster_06_9610	2017-03-03 22:27:00.641265	备节点	运行中	PostgreSQL 9.6.1 on	192.168.1.6	22
3	users_cluster_10_9610	2017-02-22 09:24:31.400303	主节点	运行中	PostgreSQL 9.6.1 on	192.168.1.10	22
4	users_cluster_10_9410	2017-03-04 11:32:36.324464	主节点	运行中	PostgreSQL 9.4.1 on	192.168.1.10	22
5	users_cluster_12_9410	2017-02-24 11:09:50.06416	普通节点	运行中	PostgreSQL 9.4.1 on	192.168.1.12	22

2、增加要维护的节点资料

Postgresql集群管理器

[+ 增加节点](#)
[修改资料](#)
[删除节点](#)
[参数配置](#)
[服务管理](#)
[VIP绑定](#)
[备机唤醒](#)
[主备切换](#)
[修改登录密码](#)
[退出](#)

	节点名称	建立时间	节点类别	运行状态	版本信息	主机名或IP	SSH
1	users_cluster_10-9620	2017-03-11 12:24:40.794889	备节点	运行中	PostgreSQL 9.6.1 on	192.168.1.10	22
2	users_cluster_06_9610	2017-03-03 22:27:00.641265	备节点	运行中	PostgreSQL 9.6.1 on	192.168.1.6	22
3	users_cluster_10_9610	2017-02-22 09:24:31.400303	主节点	运行中	PostgreSQL 9.6.1 on	192.168.1.10	22
4	users_cluster_10_9410	2017-03-04 11:32:36.324464	主节点	运行中	PostgreSQL 9.4.1 on	192.168.1.10	22
5	users_cluster_12_9410	2017-02-24 11:09:50.06416	普通节点	运行中	PostgreSQL 9.4.1 on	192.168.1.12	22

点击“增加节点”,弹出增加节点编辑窗口

100 第 1 共 1 页 显示 1 到 5, 共 5 记录

“编辑窗口” 如下所示

增加一个节点资料

节点编号: 系统自动编号

节点名称:

主机名或IP:

SSH端口号: 22

SSH认证方式: 密钥登陆 (key)

SSH用户名: postgres

SSH登录密码:

PG服务端程序路径: /usr/local/pgsql/bin/

PGDATA所在路径: /home/postgres/data/

访问日志保存路径:

PG服务端口号: 5432

连接数据库名称: template1

数据库用户名: postgres

数据库用户密码:

做为主节点绑定VIP:

VIP绑定设备号: 如eth0:1

做为备节点绑定VIP:

VIP绑定设备号: 如eth0:1

绑定网卡登录认证方式: 密钥登陆 (key)

绑定网卡操作用户:

绑定网卡操作密码:

备注:

保存 取消

操作说明:

- 1、要维护的节点如果非主备节点，则一般都不需要配置 vip 相关参数。
- 2、VIP 绑定和解绑“设备号”如果不清楚请问你们的 sa,千万别配置错了，用命令 `ip a` 可查询“设备号”。
- 3、编辑完成后按“保存”即可保存资料。

3、编辑节点资料

Postgresql集群管理器

[+ 增加节点](#)
[✎ 修改资料](#)
[- 删除节点](#)
[⚙ 参数配置](#)
[🖨 服务管理](#)
[📍 VIP绑定](#)
[💤 备机唤醒](#)
[🔄 主备切换](#)
[🔑 修改登录密码](#)
[🚪 退出](#)

	<input type="checkbox"/>	节点名称	建立时间	节点类别	运行状态	版本信息	主机名或IP	SSH
1	<input type="checkbox"/>	users_cluster_10_9620	2017-03-11 12:24:40.794889	备节点	运行中	PostgreSQL 9.6.1 on	192.168.1.10	22
2	<input type="checkbox"/>	users_cluster_06_9610	2017-03-03 22:27:00.641265	备节点	运行中	PostgreSQL 9.6.1 on	192.168.1.6	22
3	<input type="checkbox"/>	users_cluster_10_9610	2017-02-22 09:24:31.400303	主节点	运行中	PostgreSQL 9.6.1 on	192.168.1.10	22
4	<input checked="" type="checkbox"/>	users_cluster_10_9410	2017-03-04 11:32:36.324464	主节点	运行中	PostgreSQL 9.4.1 on	192.168.1.10	22
5	<input type="checkbox"/>	users_cluster_12_9410	2017-02-24 11:09:50.06416	普通节点	运行中	PostgreSQL 9.4.1 on	192.168.1.12	22

1: 选择要维护的记录

2: 点击“修改资料”，弹出修改窗口

100 第 1 共1页 显示1到5,共5记录

说明：节点资料维护窗口跟新增节点资料一致，说明请参考上面的“增加节点”。

4、删除节点资料

Postgresql集群管理器

[+ 增加节点](#)
[✎ 修改资料](#)
[- 删除节点](#)
[⚙ 参数配置](#)
[🖨 服务管理](#)
[📍 VIP绑定](#)
[💤 备机唤醒](#)
[🔄 主备切换](#)
[🔑 修改登录密码](#)
[🚪 退出](#)

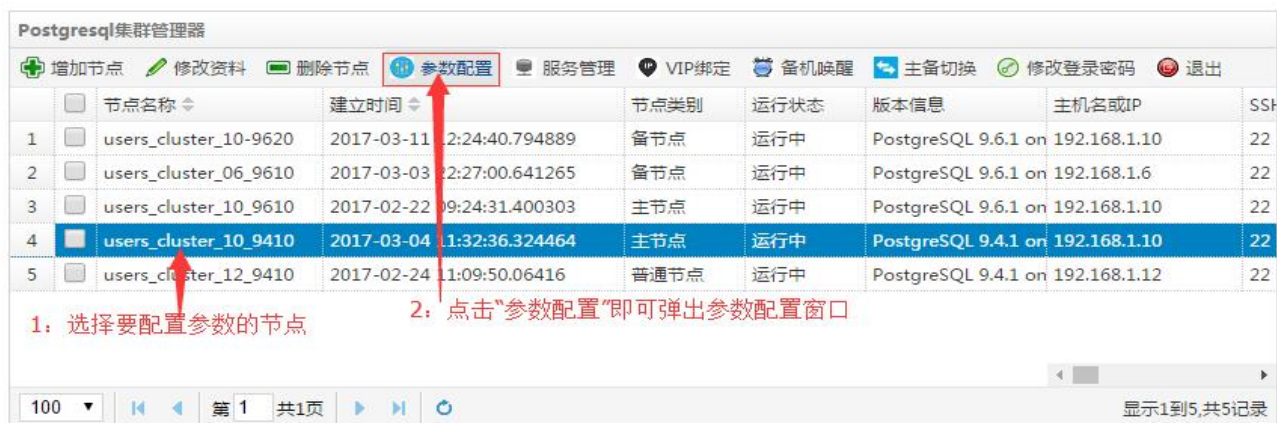
	<input type="checkbox"/>	节点名称	建立时间	节点类别	运行状态	版本信息	主机名或IP	SSH
1	<input type="checkbox"/>	users_cluster_10_9620	2017-03-11 12:24:40.794889	备节点	运行中	PostgreSQL 9.6.1 on	192.168.1.10	22
2	<input type="checkbox"/>	users_cluster_06_9610	2017-03-03 22:27:00.641265	备节点	运行中	PostgreSQL 9.6.1 on	192.168.1.6	22
3	<input type="checkbox"/>	users_cluster_10_9610	2017-02-22 09:24:31.400303	主节点	运行中	PostgreSQL 9.6.1 on	192.168.1.10	22
4	<input checked="" type="checkbox"/>	users_cluster_10_9410	2017-03-04 11:32:36.324464	主节点	运行中	PostgreSQL 9.4.1 on	192.168.1.10	22
5	<input type="checkbox"/>	users_cluster_12_9410	2017-02-24 11:09:50.06416	普通节点	运行中	PostgreSQL 9.4.1 on	192.168.1.12	22

1: 选择要删除的记录

2: 点击删除节点，在弹出的确认窗口按“确定”即可

100 第 1 共1页 显示1到5,共5记录

5、参数配置



“参数配置”窗口如下所示

操作说明:

- 1、配置文件有 postgresql.conf, pg_hba.conf, recovery.conf (备机上才有这个项目)。
- 2、修改好参数值后可以分别保存, 保存&reload, 保存&restart。
- 3、执行结果会显示在“执行结果”输出显示框中。
- 4、选择“历史备份”可以导入先前保存的版本。
- 5、选择“模板文件”可以导入已经准备好的标准模板文件。
- 6、“历史备份”, “模板文件”下拉选择框支持删除不需要的版本或模板文件。



6、节点服务管理

Postgresql集群管理器

[增加节点](#)
[修改资料](#)
[删除节点](#)
[参数配置](#)
[服务管理](#)
[VIP绑定](#)
[备机唤醒](#)
[主备切换](#)
[修改登录密码](#)
[退出](#)

	<input type="checkbox"/>	节点名称	建立时间	节点类别	运行状态	版本信息	主机名或IP	SSH
1	<input type="checkbox"/>	users_cluster_10_9620	2017-03-11 12:24:40.794889	备节点	运行中	PostgreSQL 9.6.1 on	192.168.1.10	22
2	<input type="checkbox"/>	users_cluster_06_9610	2017-03-03 22:27:00.641265	备节点	运行中	PostgreSQL 9.6.1 on	192.168.1.6	22
3	<input type="checkbox"/>	users_cluster_10_9610	2017-02-22 09:24:31.400303	主节点	运行中	PostgreSQL 9.6.1 on	192.168.1.10	22
4	<input checked="" type="checkbox"/>	users_cluster_10_9410	2017-03-04 11:32:36.324464	主节点	运行中	PostgreSQL 9.4.1 on	192.168.1.10	22
5	<input type="checkbox"/>	users_cluster_12_9410	2017-02-24 11:09:50.06416	普通节点	运行中	PostgreSQL 9.4.1 on	192.168.1.12	22

1: 选择要执行“服务管理”的节点

2: 点击“服务管理”即可弹出服务管理窗口

100 第 1 共 1 页 显示1到5,共5记录

“服务管理”窗口如下所示

服务管理

节点编号: 1 节点名称: users_cluster_10_9610

节点类别: 主节点 运行状态: 运行中

版本号: PostgreSQL 9.6.1 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-4), 64-bit

主机名或IP: 192.168.1.10 SSH端口号: 22

SSH认证方式: 密钥登陆 (key)

SSH用户名: postgres SSH登录密码:

PG服务端程序路径: /usr/local/pgsql9.6.1/bin/

PGDATA所在路径: /home/postgres/data9.6.1/

访问日志保存路径:

关闭模式: fast 选择stop/restart的执行模式

执行命令: /usr/local/pgsql9.6.1/bin/pg_ctl status -D /home/postgres/data9.6.1/

执行结果: pg_ctl: server is running (PID: 30184)
/usr/local/pgsql9.6.1/bin/postgres "-D" "/home/postgres/data9.6.1"

显示执行返回的信息

功能按钮区

start stop restart reload status 取消

操作说明:

- 1、关闭模式只能 stop/restart 操作有作用。
- 2、执行返回的结果显示在“执行结果”显示框。

7、节点 VIP 管理

节点 VIP 管理功能主要是对运行 postgresql 服务的机器网卡进行绑定或解绑对外提供服务的 IP 地址。

Postgresql集群管理器

[增加节点](#)
[修改资料](#)
[删除节点](#)
[参数配置](#)
[服务管理](#)
[VIP绑定](#)
[备机唤醒](#)
[主备切换](#)
[修改登录密码](#)
[退出](#)

	节点名称	建立时间	节点类别	运行状态	版本信息	主机名或IP	SSH
1	users_cluster_10-9620	2017-03-11 12:24:40.794889	备节点	运行中	PostgreSQL 9.6.1 on	192.168.1.10	22
2	users_cluster_10_9610	2017-02-22 09:24:31.400303	主节点	运行中	PostgreSQL 9.6.1 on	192.168.1.10	22
3	users_cluster_10_9410	2017-03-04 11:32:36.324464	主节点	运行中	PostgreSQL 9.4.1 on	192.168.1.10	22
4	users_cluster_12_9410	2017-02-24 11:09:50.06416	普通节点	运行中	PostgreSQL 9.4.1 on	192.168.1.12	22
5	users_cluster_06_9610	2017-03-03 22:27:00.641265	备节点	运行中	PostgreSQL 9.6.1 on	192.168.1.6	22

1: 选择要进行“VIP绑定”或解绑的节点 2: 点击“VIP绑定”即可弹出VIP绑定管理窗口

100 第 1 共 1 页 显示1到5,共5记录

如果前面编辑节点资料时有录入 VIP 绑定相关参数，这里会根据节点的类型自动带出，解绑和绑定的 IP 时系统会自动检查是否存在 IP 冲突情况。“IP 绑定详情”会显示当前节点机器的 IP 绑定情况。

VIP绑定管理

节点编号: 1 节点名称: users_cluster_10_9610
 节点类别: 主节点 运行状态: 运行中
 版本号: PostgreSQL 9.6.1 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-4), 64-bit
 主机名或IP: 192.168.1.10 SSH端口号: 22
 PG服务端程序路径: /usr/local/pgsql9.6.1/bin/
 PGDATA所在路径: /home/postgres/data9.6.1/

VIP地址: 网卡设备号: 如eth0:1
 绑定网卡登录认证方式: 密钥登陆 (key) 要绑定或解绑的参数在这里设置
 绑定操作用户名: 操作用户密码:

IP绑定详情

```

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
     valid_lft forever preferred_lft forever
2: em1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP qlen 1000
   link/ether c8:1f:66:c2:c8:5b brd ff:ff:ff:ff:ff:ff
   inet 192.168.1.10/24 brd 192.168.1.255 scope global em1
     valid_lft forever preferred_lft forever
   inet 192.168.1.11/24 brd 192.168.1.255 scope global secondary em1:1
     valid_lft forever preferred_lft forever
  
```

这里显示节点所在机器的ip绑定情况

执行结果: 执行返回的信息在这里显示

操作说明: 填写要解绑或绑定IP和对应的网卡设备号,再按“绑定VIP”或“解绑VIP”即可.Linux下绑定vip一般需要root权限,不清楚网卡设备号请问问SA,千万不能乱操作

功能按钮区: 绑定VIP 解绑VIP 刷新ip绑定详情 取消

操作说明:

- 1、输入要绑定的“VIP 地址”，“网卡设备号”，选择“绑定网卡登录认证方式”，“绑定操作用户名”，如果“绑定网卡登录认证方式”值为密码登录则需要录入“操作用户密码”。

- 2、点击“绑定 VIP”或“解绑 VIP”。
- 3、执行返回的结果显示在“执行结果”显示框。
- 4、点击“刷新 IP 绑定详情”在“IP 绑定详情”显示该节点所在机器的 IP 绑定情况。

8、备机唤醒

Postgresql集群管理器

[+ 增加节点](#)
[✎ 修改资料](#)
[🗑 删除节点](#)
[⚙ 参数配置](#)
[🖨 服务管理](#)
[📍 VIP绑定](#)
[🔔 备机唤醒](#)
[🔄 主备切换](#)
[🔑 修改登录密码](#)
[🚪 退出](#)

	<input type="checkbox"/>	节点名称	建立时间	节点类别	运行状态	版本信息	主机名或IP	SSH
1	<input type="checkbox"/>	users_cluster_10-9620	2017-03-11 12:24:40.794889	备节点	运行中	PostgreSQL 9.6.1 on	192.168.1.10	22
2	<input type="checkbox"/>	users_cluster_10_9610	2017-02-22 09:24:31.400303	主节点	运行中	PostgreSQL 9.6.1 on	192.168.1.10	22
3	<input type="checkbox"/>	users_cluster_10_9410	2017-03-04 11:32:36.324464	主节点	运行中	PostgreSQL 9.4.1 on	192.168.1.10	22
4	<input type="checkbox"/>	users_cluster_12_9410	2017-02-24 11:09:50.06416	普通节点	运行中	PostgreSQL 9.4.1 on	192.168.1.12	22
5	<input checked="" type="checkbox"/>	users_cluster_06_9610	2017-03-03 22:27:00.641265	备节点	运行中	PostgreSQL 9.6.1 on	192.168.1.6	22

2：点击“备机唤醒”即可弹出备机唤醒管理窗口

1：选择要进行“备机唤醒”的节点

100 第 1 共 1 页 显示 1 到 5, 共 5 记录

“备机唤醒”窗口如下所示

备机唤醒管理

节点编号: 2 节点名称: users_cluster_06_9610
 节点类别: 备节点 运行状态: 运行中
 版本号: PostgreSQL 9.6.1 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.4.7 20120313 (Red Hat 4.4.7-11), 64-bit
 主机名或IP: 192.168.1.6 SSH端口号: 22
 PG服务端程序路径: /usr/local/pgsql9.6.1/bin/
 PGDATA所在路径: /home/postgres/data9.6.1/

唤醒后绑定VIP地址: 绑定网卡设备号: 如eth0:1
 绑定网卡登录认证方式: 密钥登陆 (key) 唤醒成功后如果需要绑定VIP,则在这里设置相关参数
 绑定操作用户名: 操作用户密码:

唤醒后执行VACUUM: ☒ 打开则在备机唤醒成功后执行VACUUM -- 建议打开

IP绑定详情:


```

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
2: em1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP qlen 1000
    link/ether c8:1f:66:c5:14:35 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.6/24 brd 192.168.1.255 scope global em1
    inet 192.168.1.9/24 brd 192.168.1.255 scope global secondary em1:0
    inet 192.168.1.12/24 brd 192.168.1.255 scope global secondary em1:12
    inet 192.168.1.13/24 brd 192.168.1.255 scope global secondary em1:1
3: em2: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state UNKNOWN qlen 1000
    link/ether 08:00:27:00:00:00 brd ff:ff:ff:ff:ff:ff
    
```

 这里显示该节点所在机器的IP绑定情况

执行结果: 执行返回的信息在这里显示

操作说明: 备机唤醒后,如果节点为同步复制模式并且备机连接上来,则系统会修改运行参数把节点的同步模式修改为异步模式。

功能按钮区

[执行](#)
[刷新ip绑定详情](#)
[取消](#)

操作说明：

- 1、如果备机唤醒成功后需要绑定 VIP 供业务连接使用，则需要录入“唤醒后绑定 VIP 地址”，“绑定网卡设备号”，选择“绑定网卡登录认证方式”，“绑定操作用户名”，如果“绑定网卡登录认证方式”值为密码登录则需要录入“操作用户密码”。程序会自动检测要绑定的 IP 是否已经被其它机器占用。
- 2、“唤醒后执行 VACUUM”是一个执行开关，打开表示备机唤醒成功后需要执行垃圾回收操作，默认是打开，备机唤醒后在提供服务前建议执行这个操作。
- 3、配置好相关参数后按“执行”按钮即可唤醒备机，如果执行出错，错误信息将显示在“执行结果”显示框中。
- 4、点击“刷新 IP 绑定详情”在“IP 绑定详情”显示该节点所在机器的 IP 绑定情况。
- 5、唤醒成功后系统会自动检查节点是否为“同步复制模式”和是否有“同步备机”连接上来，如果是“同步复制模式”并且没有“同步备机”连接上来，程序会主动把节点降级为“异步复制”模式。

9、主备切换

The screenshot shows the 'Postgresql集群管理器' (PostgreSQL Cluster Manager) interface. At the top, there is a navigation bar with buttons: '增加节点' (Add Node), '修改资料' (Modify Info), '删除节点' (Delete Node), '参数配置' (Parameter Config), '服务管理' (Service Management), 'VIP绑定' (VIP Binding), '备机唤醒' (Slave Wakeup), '主备切换' (Main-Slave Switch), '修改登录密码' (Modify Login Password), and '退出' (Exit). The '主备切换' button is highlighted with a red box and an arrow pointing to it from the annotation '2: 单击“主备切换”按钮即可弹出主备切换管理窗口'.

Below the navigation bar is a table with the following columns: '节点名称' (Node Name), '建立时间' (Creation Time), '节点类别' (Node Type), '运行状态' (Running Status), '版本信息' (Version Info), '主机名或IP' (Hostname or IP), and 'SSH端' (SSH Port). The table contains 5 rows of data. The second row, 'users_cluster_10_9610', is selected, indicated by a red box and an arrow from the annotation '1: 选择要切换的主备节点'.

	节点名称	建立时间	节点类别	运行状态	版本信息	主机名或IP	SSH端
1	users_cluster_10-9620	2017-03-11 12:24:40.794889	备节点	运行中	PostgreSQL 9.6.1 on	192.168.1.10	22
2	users_cluster_10_9610	2017-02-22 09:24:31.400303	主节点	运行中	PostgreSQL 9.6.1 on	192.168.1.10	22
3	users_cluster_10_9410	2017-03-04 11:32:36.324464	主节点	运行中	PostgreSQL 9.4.1 on	192.168.1.10	22
4	users_cluster_12_9410	2017-02-24 11:09:50.06416	普通节点	运行中	PostgreSQL 9.4.1 on	192.168.1.12	22
5	users_cluster_06_9610	2017-03-03 22:27:00.641265	备节点	运行中	PostgreSQL 9.6.1 on	192.168.1.6	22

At the bottom of the interface, there is a pagination bar showing '100' items, '第 1 共 1 页' (Page 1 of 1), and a '显示1到5,共5记录' (Display 1 to 5, total 5 records) indicator.

“主备切换”框窗口如下所示

主备切换管理

主节点 (users_cluster_10_9610)

备节点 (users_cluster_06_9610)

切换参数配置

主节点信息显示框

备节点信息显示框

主备切换参数配置框

节点编号	1	节点名称	users_cluster_10_9610
节点类别	主节点	运行状态	运行中
版本号	PostgreSQL 9.6.1 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-4), 64-bit		
主机名或IP	192.168.1.10	SSH端口号	22
SSH认证方式	密钥登陆 (key)		
SSH用户名	postgres	SSH登录密码	
PG服务端程序路径	/usr/local/pgsql9.6.1/bin/		
PGDATA所在路径	/home/postgres/data9.6.1/		
PG服务端口号	9610	连接数据库名称	template1
数据库用户名	postgres	数据库用户密码
IP绑定详情	<pre> link/ether c8:1f:66:c2:c8:5b brd ff:ff:ff:ff:ff:ff inet 192.168.1.10/24 brd 192.168.1.255 scope global em1 valid_lft forever preferred_lft forever inet 192.168.1.11/24 brd 192.168.1.255 scope global secondary em1:1 valid_lft forever preferred_lft forever 3: em2: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN qlen 1000 link/ether c8:1f:66:c2:c8:5c brd ff:ff:ff:ff:ff:ff 4: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN link/ether 52:54:00:66:89:12 brd ff:ff:ff:ff:ff:ff inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0 valid_lft forever preferred_lft forever 5: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master virbr0 state DOWN qlen 500 link/ether 52:54:00:66:89:12 brd ff:ff:ff:ff:ff:ff </pre>		

功能按键区

一键切换

刷新ip绑定详情

取消

操作说明:

- 1、主备切换需要同时勾选两个节点记录，两个节点必需是一主一备。
- 2、要切换的两个节点当前状态必需是处于“运行中”。
- 3、系统会判断两个节点是否为主备关系。
- 4、如果需要解绑和绑定 VIP 的话，则需要配置“主节点切为备节点参数”和“备节点切为主节点参数”。
- 5、程序会自动检测要绑定的 IP 是否已经被其它机器占用。
- 5、配置完参数后，按“一键切换”即可完成主备切换。
- 6、如果如果执行出错，错误信息将显示在“执行结果”显示框中。
- 7、点击“刷新 IP 绑定详情”在主备两个页框的“IP 绑定详情”显示节点所在机器的 IP 绑定情况。

“切换参数配置”框如下图所示

主备切换管理

主节点 (users_cluster_10_9610)

备节点 (users_cluster_06_9610)

切换参数配置

复制参数 (recovery.conf)

```
archive_cleanup_command = '/usr/local/pgsql9.6.1/bin/pg_archivecleanup /home/postgres/data9.6.1/pg_xlog %r'
standby_mode = 'on'
primary_conninfo = 'host=192.168.1.6 port=9610 user=postgres password=pgsql'
recovery_target_timeline = 'latest'
```

主备复制参数自动生成，可以再次修改

主节点切为备节点参数

切换前解绑vip

192.168.1.11

解绑网卡设备号

em1:1

切换后绑定vip

192.168.1.13

绑定网卡设备号

em1:1

绑定网卡登录认证方式

密钥登陆 (key)

绑定操作用户

root

操作用户密码

当前主节点切换前后需要使用的参数在这里配置

备节点切为主节点参数

切换前解绑vip

192.168.1.13

解绑网卡设备号

em1:1

切换后绑定vip

192.168.1.11

绑定网卡设备号

em1:1

绑定网卡登录认证方式

密码登陆 (password)

绑定操作用户

root

操作用户密码

.....

当前备节点切换前后需要使用的参数在这里配置

执行结果

执行返回的信息在这里显示

切换说明

除复制参数 (recovery.conf) 外，vip绑定视需要填写，不填写不执行vip绑定功能，Linux下绑定vip一般需要root权限。

配置好参数扣按“一键切换”搞定主备切换工作

一键切换

刷新ip绑定详情

取消

10、巡检报告管理

Postgresql集群管理器

2、点击巡检报告弹出管理窗口

增加节点

修改资料

删除节点

参数配置

服务管理

VIP绑定

备机唤醒

主备切换

巡检报告

修改登录密码

退出

	节点名称	建立时间	节点类别	运行状态	版本信息	主机名或IP	SSH端口号
1	users_cluster_10_9620	2017-04-11 22:00:20.451669	备节点	运行中	PostgreSQL 9.6.1 on	192.168.1.10	22
2	dywl_test_12_5433	2017-04-11 22:00:20.451669	普通节点	运行中	PostgreSQL 9.4.1 on	192.168.1.12	22
3	dywl_develop_10_5432	2017-04-07 15:23:15.157256	主节点	运行中	PostgreSQL 9.4.1 on	192.168.1.10	22
4	users_cluster_06_9610	2017-04-06 13:42:14.457722	备节点	运行中	PostgreSQL 9.6.1 on	192.168.1.6	22
5	users_cluster_10_9610	2017-04-06 13:42:13.197434	主节点	运行中	PostgreSQL 9.6.1 on	192.168.1.10	22

100

第 1 共 1 页

显示1到5,共5记录

巡检报告管理窗口如下所示

已经生成的“巡检报告”列表
统计已完成值为“否”表示该巡检报告还在后台统计表的记录数

报告名称	生成时间	统计已完成	操作员
20170808-01	2017-08-08 10:3	否	admin
20170807-02	2017-08-07 23:2	是	admin
20170807-01	2017-08-07 23:2	是	admin

ID号	项目	值
1	274	192.168.0.231
2	275	postgres
3	276	/usr/local/pgsql9.6.11/bin/
4	277	/home/postgres/data9.6.1/
5	278	/home/postgres/data9.6.1/pg_log/
6	279	5432
7	280	PostgreSQL 9.6.1 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.4.7 20
8	281	主节点
9	282	PRC
10	283	256MB
11	284	on
12	285	csvlog
13	286	on
14	287	PRC
15	288	277 MB

功能按钮区: 创建, 修改, 删除, 导出, 取消

操作说明:

- 1、点击左边的巡检报告列表即可在右边显示该报告的明细
- 2、点击右边的报告的明细页框标题即可查看各个统计对象的明细数据
- 3、点击按钮“创建”，弹出创建对话框，录入参数后按确定即可生成巡检报告（注意：比较消耗资源，建议在系统空闲时生成）
- 4、点击按钮“修改”，可以修改巡检报告名称。
- 5、点击按钮“删除”，可以删除选择的巡检报告。
- 6、点击按钮“导出”，可以导出 excel 文件，内容为巡检报告明细数据

创建巡检报告对话框

填写巡检报告名称: pgcluster20170808日巡检报告表 这里录入巡检报告名称, 怎样好记就怎样录入。

选择统计的数据库: pgcluster - postgresql集群管理器 选择要统计的数据库, 默认统计所有数据库。

表行数统计进程数: 2 配置统计数据表记录数的并发进程数, 0表示不统计, 注: 统计表记录数非常消耗资源。

是否统计系统对象: 不统计 选择是否统计系统表的数据

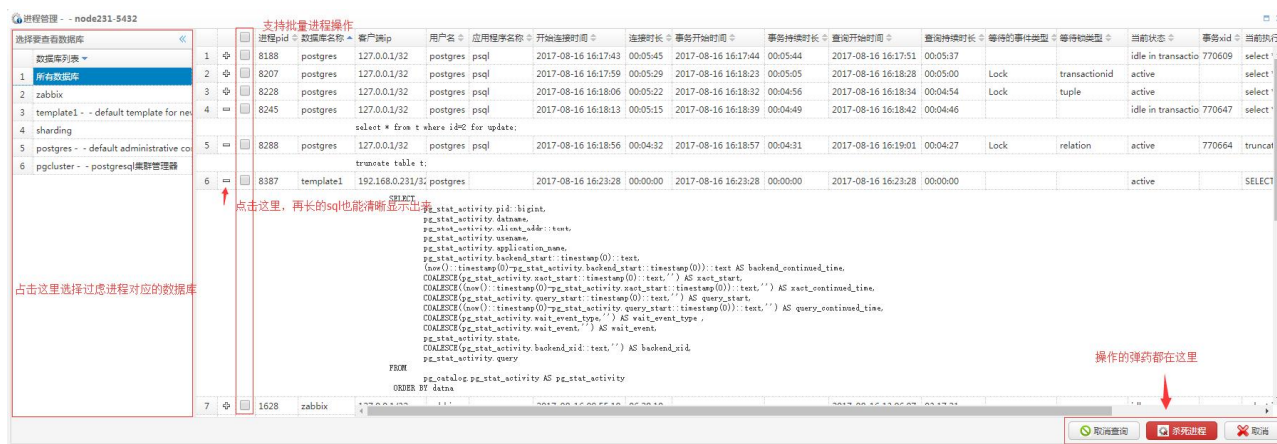
确认创建, 弹出“创建巡检报告”对话框

统计表记录数变更成异步统计, 所以现在提示生成报告成功后, 表记录数有可能还没统计完成, 特别是大表的统计比较占用时间。只有当你看到巡检报告列表中的“统计已完成”值为“是”, 这时所有统计工作才完成。

11、管理工具——进程管理



进程管理窗口如下



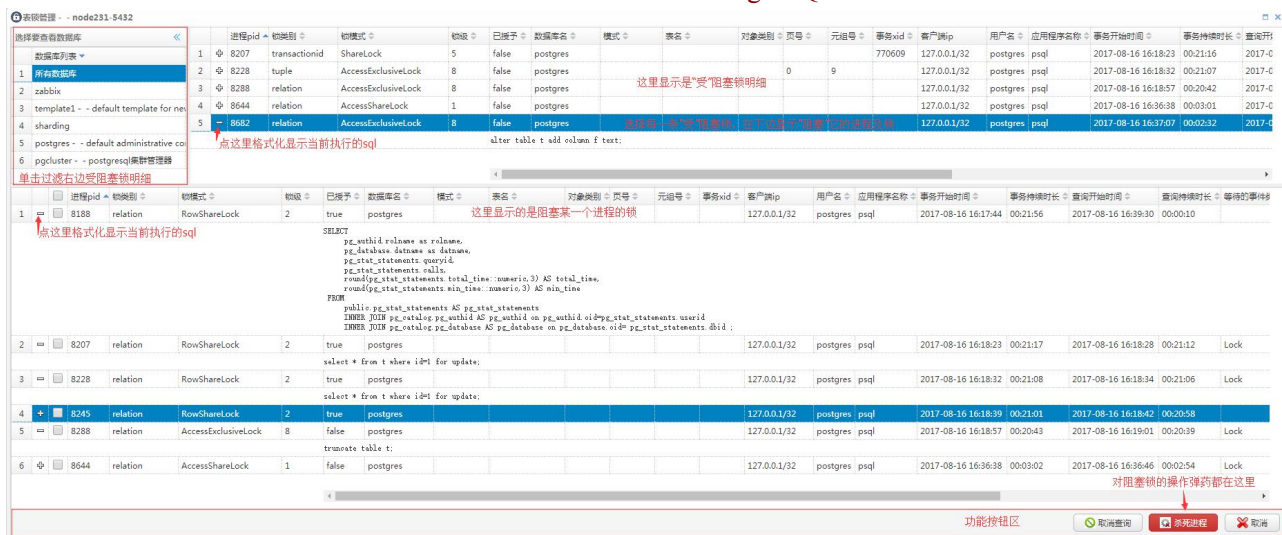
操作说明：

- 1、点击左边的数据库列表，在右边的进程列表只显示该数据库的连接进程
- 2、单击右边列表“+”号可以格式化显示进程当前执行的Sql语句
- 3、选择要操作的进程（支持多选），按“取消查询”可以取消掉进程当前正在执行的查询，其对应执行的操作就是 `pg_cancel_backend(pid)`
- 4、选择要操作的进程（支持多选），按“杀死进程”可以杀掉选择的进程，其对应执行的操作就是 `pg_terminate_backend(pid)`

12、管理工具——表锁管理



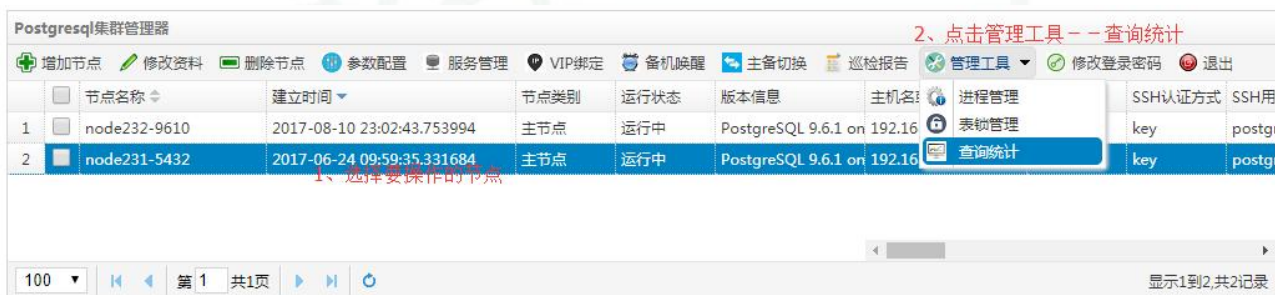
表锁管理窗口如下



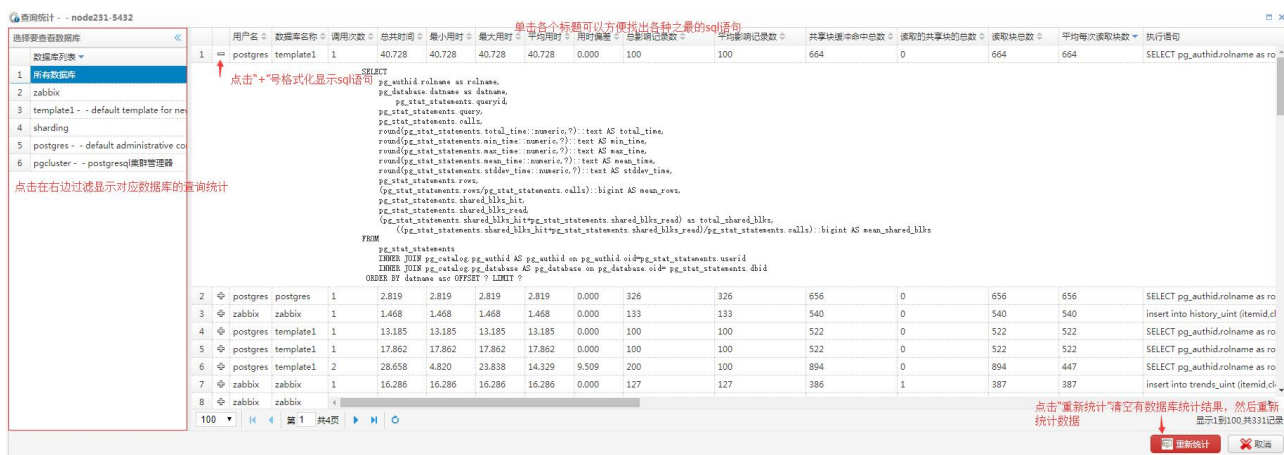
操作说明:

- 1、点击左上边的数据库列表，在右边显示该数据库的相关的“受”阻塞锁列表
- 2、点击右上边“受”阻塞锁列表，可在下边显示“阻塞”它的进程列表
- 3、单击列表“+”号可以格式化显示进程当前执行的Sql 语句
- 4、选择要操作的进程（支持多选），按“取消查询”可以取消掉进程当前正在执行的查询，其对应执行的操作就是 `pg_cancel_backend(pid)`
- 5、选择要操作的进程（支持多选），按“杀死进程”可以杀掉选择的进程，其对应执行的操作就是 `pg_terminate_backend(pid)`

13、管理工具——查询统计



查询统计窗口如下

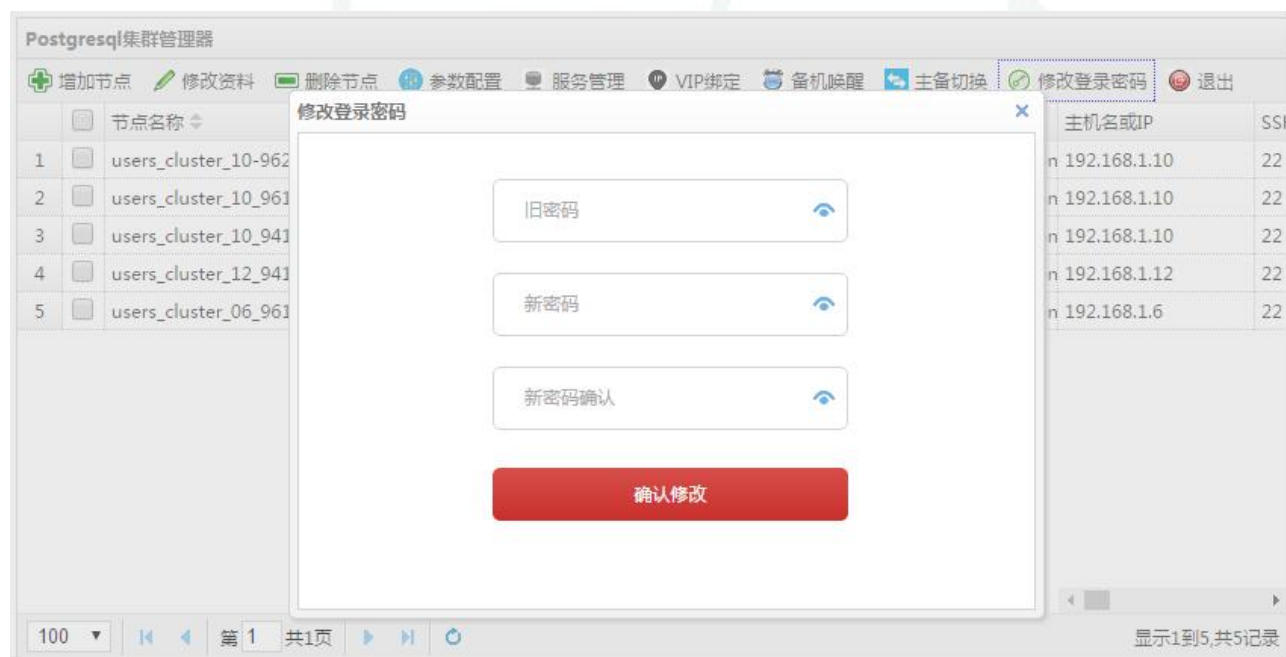


操作说明：

- 1、点击左上边的数据库列表，在右上边显示该数据库相关的 Sql 统计记录
- 2、单击列表“+”号可以格式化显示统计的 sql 语句

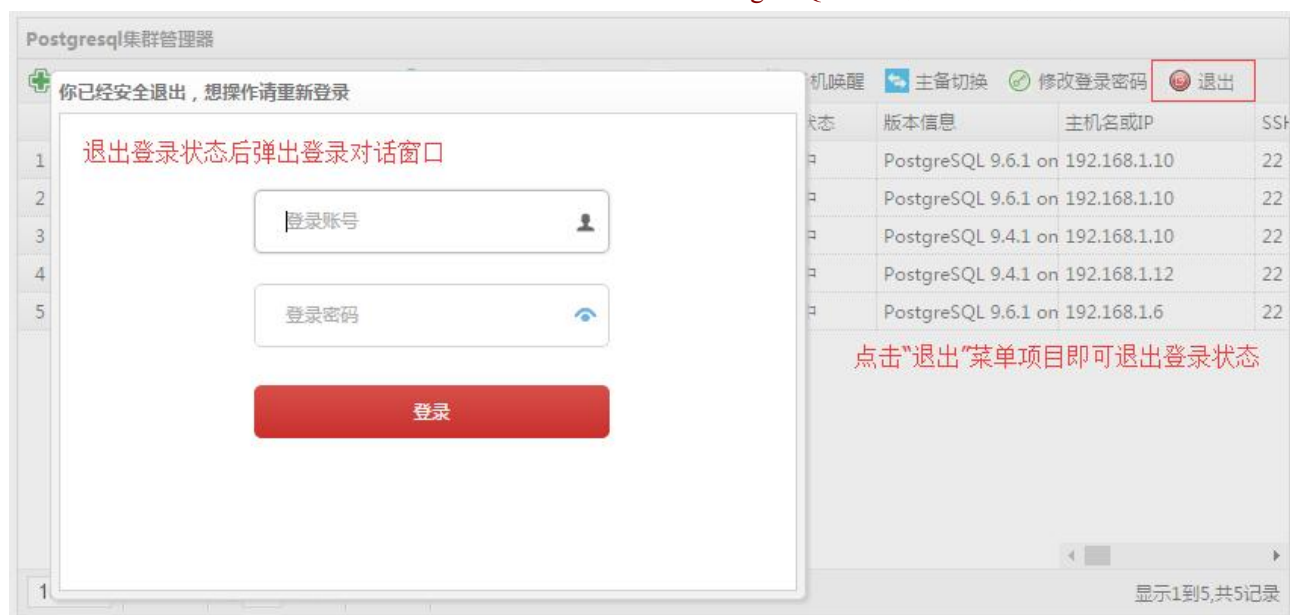
14、修改登录密码

下面是修改登录密码窗口

**操作说明：**

- 1、输入旧的登录密码。
- 2、新的密码需要连续输入两次。
- 3、录入完成后按“确认修改”即可完成登录密码的修改。

- 退出登录状态



退出后系统返回到登录页面。

六、接下来开发计划

调整前后端代码，各个模块解耦独立出来，方便后期多人工作协同

社区网站

PostgreSQL 全球社区网: <http://www.postgresql.org>

PostgreSQL 中国社区网: <http://www.postgres.cn>

社区 QQ 群

技术群#1: 5276420

技术群#2: 3336901

文档翻译群: 309292849

二维码



官方微信公众号



官方微博



阿弟@postgresql-微信号

