

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，当前主要功能是实现对 postgresql 服务进行管理，主备切换进行管理；系统支持多用户，操作认证；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。
- 单一节点服务 start、stop、restart、reload 及显示服务状态。
- 主备节点一键切换。

二、部署环境

Ip: 192.168.1.10

os: centos 7.0

golang: go version go1.7.4 linux/amd64

Postgresql: 9.6.1

三、Pgclusteradmin 部署方法

- 安装 golang

```
[root@ad ~]# yum install golang-1.7.4-1.el6.x86_64.rpm
```

```
[root@ad ~]# yum install golang-src-1.7.4-1.el6.noarch.rpm
```

```
[root@ad ~]# yum install golang-bin-1.7.4-1.el6.x86_64.rpm
```

- 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 = 'stderr'
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。

- 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
```

——导入下面数据表及数据

——节点资料表

```
create table nodes
(
    id serial not null unique,
    node_name text not null unique,
    createtime timestamp not null default now(),
    host text not null,
    ssh_port integer not null,
    ssh_user text not null,
    ssh_password text not null,
    pg_bin text not null,
    pg_data text not null,
    pg_log text not null default "",
    pg_port integer not null,
    pg_database text not null,
    pg_user text not null,
```

```

pg_password text not null,
master_vip text,
master_vip_networkcard text,
slave_vip text,
slave_vip_networkcard text,
bind_vip_user text,
bind_vip_password text,
remark text
);
COMMENT ON TABLE nodes IS '节点资料表';
COMMENT ON COLUMN nodes.id IS '系统编号';
COMMENT ON COLUMN nodes.node_name IS '节点名称';
COMMENT ON COLUMN nodes.createtime IS '建立时间';
COMMENT ON COLUMN nodes.host IS '主机名或 ip';
COMMENT ON COLUMN nodes.ssh_port IS 'ssh 服务端口号';
COMMENT ON COLUMN nodes.ssh_user IS 'ssh 用户';
COMMENT ON COLUMN nodes.ssh_password IS 'ssh 密码';
COMMENT ON COLUMN nodes.pg_bin IS 'pg 管理程序所在路径';
COMMENT ON COLUMN nodes.pg_data IS 'pgDATA 所在路径';
COMMENT ON COLUMN nodes.pg_log IS '用户访问日志保存路径';
COMMENT ON COLUMN nodes.pg_port IS 'pg 服务端口号';
COMMENT ON COLUMN nodes.pg_user IS 'pg 用户';
COMMENT ON COLUMN nodes.pg_password IS 'pg 密码';
COMMENT ON COLUMN nodes.master_vip IS '主节点时绑定 VIP';
COMMENT ON COLUMN nodes.master_vip_networkcard IS '主节点时绑定网卡设备号';
COMMENT ON COLUMN nodes.slave_vip IS '备节点时绑定 VIP';
COMMENT ON COLUMN nodes.slave_vip_networkcard IS '备节点时绑定网卡设备号';
COMMENT ON COLUMN nodes.bind_vip_user IS '绑定网卡操作用户';
COMMENT ON COLUMN nodes.bind_vip_password IS '绑定网卡操作密码';

```

--操作员资料表

```

CREATE TABLE users
(
    id serial not null unique,
    username text not null unique,
    password text not null
);
COMMENT ON TABLE users IS '操作员资料表';
COMMENT ON COLUMN users.id IS '系统编号';
COMMENT ON COLUMN users.username IS '登录账号';
COMMENT ON COLUMN users.password IS '登录密码 md5 值';

```

--增加一个操作员记录表

```

INSERT INTO users (username,password) values('admin',md5('admin'));

```

--操作日志表

```
CREATE TABLE log
```

```
(
    id serial not null unique,
    createtime timestamp not null default now(),
    remote_ip text,
    modlename text,
    username text,
    log_level text,
    remark text
);
```

```
COMMENT ON TABLE log IS '日志表';
```

```
COMMENT ON COLUMN log.id IS '系统编号';
```

```
COMMENT ON COLUMN log.createtime IS '访问时间';
```

```
COMMENT ON COLUMN log.remote_ip IS '访问客户端 ip 地址';
```

```
COMMENT ON COLUMN log.username IS '用户名';
```

```
COMMENT ON COLUMN log.modlename IS '模块名称';
```

```
COMMENT ON COLUMN log.log_level IS '日志级别';
```

```
COMMENT ON COLUMN log.remark IS '日志内容';
```

--存储 pg 参数文件内容多个版本和公用模板

```
CREATE TABLE parameter_bak_template
```

```
(
    id serial not null unique,
    nodeid integer not null,
    createtime timestamp not null default now(),
    username text not null,
    filename text not null,
    version text not null,
    content text not null,
    category text not null,
    remark text not null
);
```

```
COMMENT ON TABLE parameter_bak_template IS '参数文件备份或者模板表';
```

```
COMMENT ON COLUMN parameter_bak_template.id IS '系统编号';
```

```
COMMENT ON COLUMN parameter_bak_template.nodeid IS '节点 id 号';
```

```
COMMENT ON COLUMN parameter_bak_template.createtime IS '备份日期';
```

```
COMMENT ON COLUMN parameter_bak_template.username IS '操作员账号';
```

```
COMMENT ON COLUMN parameter_bak_template.filename IS '文件名称';
```

```
COMMENT ON COLUMN parameter_bak_template.version IS '版本号';
```

```
COMMENT ON COLUMN parameter_bak_template.content IS '内容';
```

```
COMMENT ON COLUMN parameter_bak_template.category IS '类别，值为 bak 或者 template';
```

COMMENT ON COLUMN parameter_bak_template.remark IS '备注';

- 下载 pgclusteradmin 所需要的 go 支持包

——ssh 支持包

```
[root@ad ~]# cd /usr/lib/golang/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/lib/golang/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 astaxie]# cd /usr/lib/golang/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.
```

四、pgclusteradmin 部署配置和访问

- 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]#
```

- 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 = "192.168.1.10" //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。

- 运行 pgclusteradmin

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

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

- 访问 pgclusteradmin

打开一个浏览器，输入 <http://192.168.1.10:10001> 即可进入管理器，192.168.1.10 换成你自己 ip 地址即可。

五、Pgclusteradmin 使用文档

- 主界面说明

Postgresql集群管理器

操作菜单项目

	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

要维护的节点列表

100 第 1 共1页 翻页/刷新/记录数统计 显示1到5,共5记录

- 增加要维护的节点资料

Postgresql集群管理器

增加节点

	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
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	users_cluster_10_9410
主机名或IP	192.168.1.10	SSH端口号	22
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	/home/postgres/data/pg_log/		
PG服务端口号	5432	连接数据库名称	template1
数据库用户名称	postgres	数据库用户密码
做为主节点绑定VIP	192.168.1.11	VIP绑定设备号	eth0:1
做为备节点绑定VIP	192.168.1.13	VIP绑定设备号	eth0:1
绑定网卡操作用户	root	绑定网卡操作密码
备注	演示使用		

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。

● 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](#)

	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 11:09:56.06416	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”。

● 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

● 参数配置

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: 选择要配置参数的节点

2: 点击“参数配置”即可弹出参数配置窗口

100 第 1 共 1 页 显示 1 到 5, 共 5 记录

“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。

参数配置

节点编号: 41 节点名称: users_cluster_10-9620

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) 4.8.5 20150623 (Red Hat 4.8.5-4), 64-bit

配置文件: postgresql.conf 这里选择要配置的参数文件

历史备份: 这里选择恢复到先前保存的版本 模板文件: 这里选择导入模板参数文件

文件内容:

```
#
# (The = is optional.) Whitespace may be used. Comments are introduced with
# # anywhere on a line. The complete list of parameter names and allowed
# values can be found in the PostgreSQL documentation.
#
# The commented-out settings shown in this file represent the default values.
# Re-commenting a setting is NOT sufficient to revert it to the default value;
# you need to reload the server. 在这里修改参数值
#
# This file is read on server startup and when the server receives a SIGHUP
# signal. If you edit the file on a running system, you have to SIGHUP the
# server for the changes to take effect, or use pg_ctl reload. Some
# parameters, which are marked below, require a server shutdown and restart to
# take effect.
#
# Any parameter can also be given as a command-line option to the server, e.g.,
# postgres -c log_connections=on. Some parameters can be changed at run time
# with the SET SQL command.
#
# Memory units: kB = kilobytes      Time units: ms = milliseconds
```

备注:

执行结果: 执行结果返回值在这里显示

功能按钮操作区: 保存 保存&reload 保存&restart 保存为模板 取消

● 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

	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

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 页 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	32	节点名称	users_cluster_10_9410
节点类别	主节点	运行状态	运行中
版本号	PostgreSQL 9.4.1 on x86_64-unknown-linux-gnu, compiled by gcc (GCC) 4.8.3 20140911 (Red Hat 4.8.3-9), 64-bit		
主机名或IP	192.168.1.10	SSH端口号	22
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	/home/postgres/data/pg_log/		
关闭模式	fast 选择执行模式, 只能stop/restart起作用		
执行命令	/usr/local/pgsql/bin/pg_ctl status -D /home/postgres/data/		
执行结果	<pre>pg_ctl: server is running (PID: 28510) /usr/local/pgsql/bin/postgres</pre> <p>这里显示执行后返回的信息</p>		

这里是功能按钮区

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。

● 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: 32 节点名称: users_cluster_10_9410

节点类别: 主节点 运行状态: 运行中

版本号: PostgreSQL 9.4.1 on x86_64-unknown-linux-gnu, compiled by gcc (GCC) 4.8.3 20140911 (Red Hat 4.8.3-9), 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/pgsql/bin/

PGDATA所在路径: /home/postgres/data/

VIP地址: 网卡设备号: 如eth0:1

绑定操作用户名: 操作用户密码:

IP绑定详情:

```

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

```

执行结果: 绑定或解绑IP执行结果显示区

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:

操作说明:

- 1、输入要绑定的“VIP 地址”，“网卡设备号”，“绑定操作用户名”，“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。

● 备机唤醒

Postgresql集群管理器

	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“备机唤醒”的节点

2: 点击“备机唤醒”即可弹出备机唤醒管理窗口

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

节点编号

30

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

绑定操作用户名

操作用户密码

唤醒后执行VACUUM

ON

唤醒成功后是否执行VACUUM开关，建议执行

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 c8:1f:66:c5:14:36 brd ff:ff:ff:ff:ff:ff
   inet 192.168.1.15/24 brd 192.168.1.255 scope global em2

```

执行结果

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”，“绑定操作用户名”，“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。

● 主备切换

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端
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 checked="" 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

1: 选择要切换的主备节点

2: 单击“主备切换”按钮即可弹出主备切换管理窗口

100 第 1 共 1 页 显示1到5,共5记录

“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](#)

主节点参数显示框: 节点编号 2, 节点类别 主节点
 备节点参数显示框: 节点编号 5, 节点类别 备节点
 主备切换参数设置框: 节点名称 users_cluster_10_9610, 运行状态 运行中

版本号: PostgreSQL 9.6.1 on x86_64-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

```

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 08:1f:66:02:08:5b 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 08:1f:66:02:08:5c 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
  
```

主备节点IP绑定情况显示区

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备主关系。
- 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'
```

主备复制参数设置区，默认系统自动生成，你可以视需求再修改

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

绑定操作用户: root 操作用户密码:

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

绑定操作用户: root 操作用户密码:

执行结果

执行失败时，在这里显示失败的原因

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。

一键切换 刷新ip绑定详情 取消

● 修改登录密码

Postgresql集群管理器

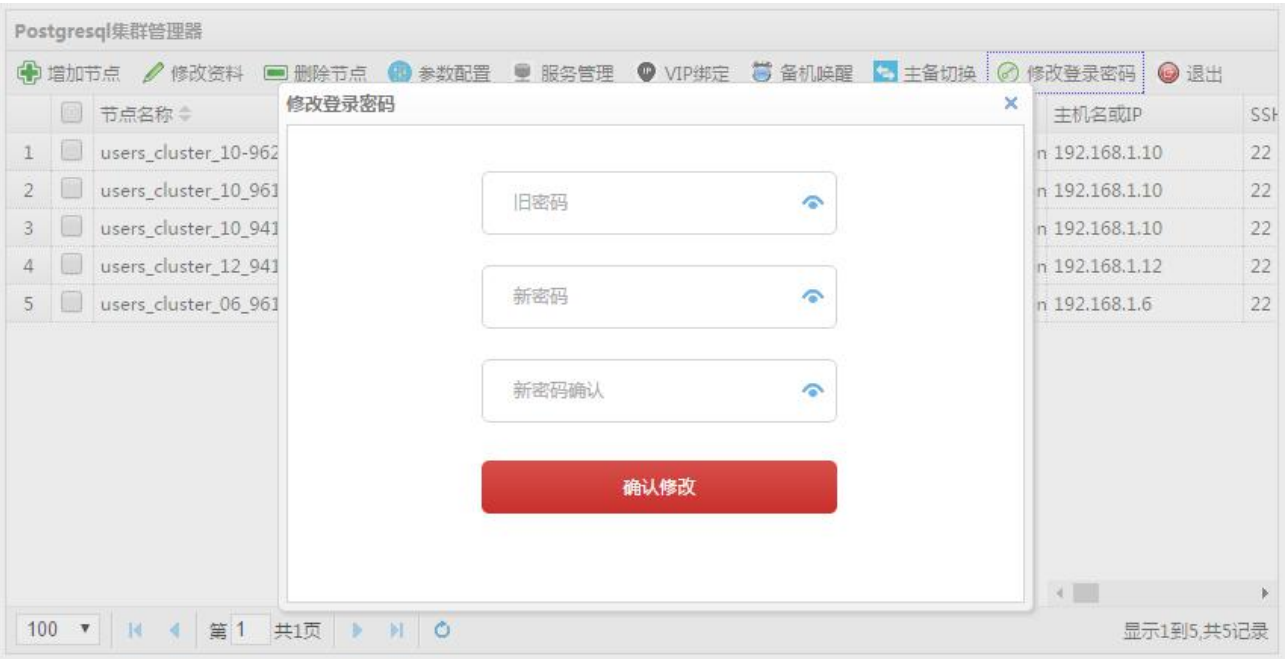
增加节点 修改资料 删除节点 参数配置 服务管理 VIP绑定 备机唤醒 主备切换 修改登录密码 退出

	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

点击“修改登录密码”，即可弹出修改登录密码窗口

100 第 1 共 1 页 显示 1 到 5, 共 5 记录

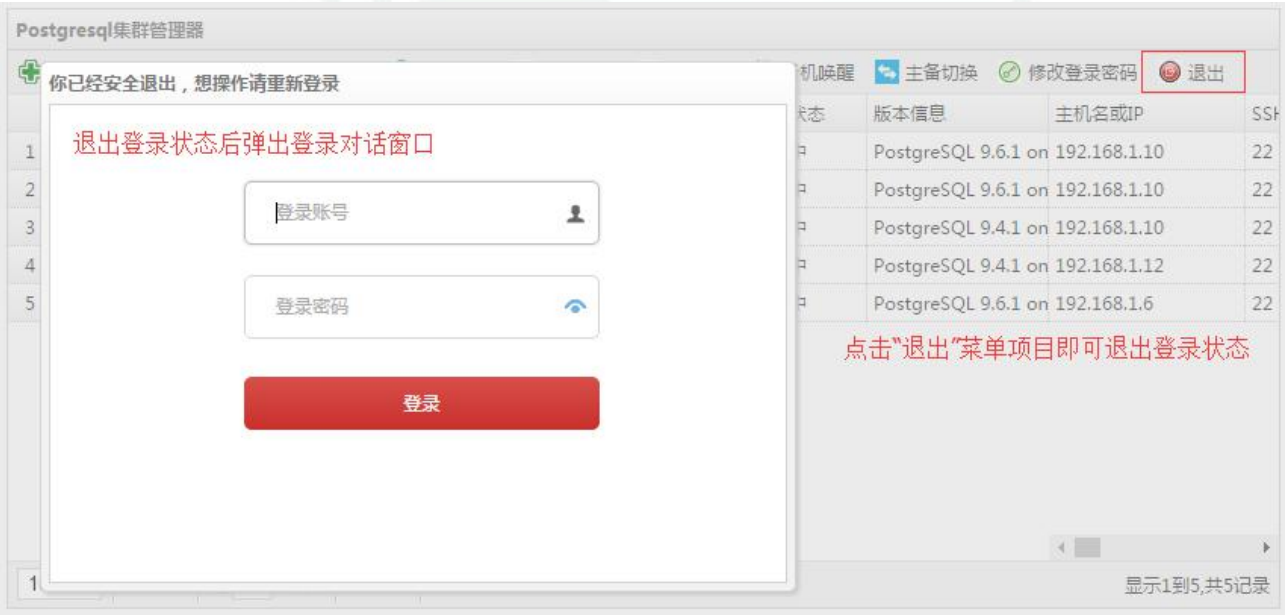
下面是修改登录密码窗口



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

短期内

- 1、增加操作员管理
- 2、增加权限管理
- 3、根据 go 的异步执行特性，优化或重写部分代码，让应用执行效率更高

长期内

- 1、数据库相关对象管理模块
- 2、数据库一些指标监控模块
- 3、数据库巡检相关模块

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