

Postgres-XL集群环境搭建

集群准备

名称	服务器	角色
gtm	10.113.75.121	gtm
node1	10.113.75.122	coordinator & datanode & gtm_proxy
node2	10.113.75.123	coordinator & datanode & gtm_proxy

所有主机添加 `hosts` 解析配置

```
# vi /etc/hosts
10.113.75.121 gtm
10.113.75.122 node1
#10.113.75.123 node2
```

使hosts生效

```
/etc/init.d/network restart
```

以下操作，对每个服务器节点都适用。关闭防火墙：

```
[root@localhost ~]# systemctl stop firewalld.service
[root@localhost ~]# systemctl disable firewalld.service
```

selinux设置:

```
[root@localhost ~]#vim /etc/selinux/config
```

设置 `SELINUX=disabled`，保存退出。

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of three two values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are
protected.
#     mls - Multi Level Security protection.
```

重启

```
reboot
```

每个节点都建立用户 `postgres`，并且建立 `.ssh` 目录，并配置相应的权限：

```
useradd postgres
passwd postgres
```

仅在gtm节点配置如下操作:

```
#在gtm节点创建key
su postgres
ssh-keygen -t rsa
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
chmod 600 .ssh

# 600 代表只有当前用户具有读、写权限，这也是安全方面的考虑。
# 权限设置为600, 否则无法登陆成功
chmod 600 authorized_keys

# 在node1上创建.ssh目录并赋值权限
su postgres
cd ~
chmod 600 .ssh

#回到gtm节点，复制key到其它节点
scp ~/.ssh/authorized_keys postgres@node1:~/.ssh/
scp ~/.ssh/authorized_keys postgres@node2:~/.ssh/
```

安装postgres-xl (每个节点都需安装)

yum源修改(非必须)

```
mv /etc/yum.repos.d/* /etc/yum.repos.d/back
wget -O /etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-7.repo
yum clean all
yum makecache
yum repolist
yum update
```

安装所需依赖包:

```
yum install -y flex bison readline-devel zlib-devel openjade docbook-style-dsssl gcc
bzip2
```

安装 postgres-xl 和 pgxc_ctl :

```
# 习惯在/usr/local下面操作
cd /usr/local
mkdir tar.gz
cd tar.gz

#下载解压( 也可以自行下载上传到/usr/local/tar.gz目录下)
wget https://www.postgres-xl.org/downloads/postgres-xl-10alpha2.tar.bz2
tar -jxvf postgres-xl-10alpha2.tar.bz2

# 若无法解压需安装解压工具
yum -y install bzip2
```

```
# 若权限不足，可对指定用户授权
chown postgres:postgres /home/postgres/pgxl
# 修改文件所有者
chown -R postgres /home/postgres/pgxl
chmod -Rf 777 /home/postgres/pgxl

#安装
./configure --prefix=/home/postgres/pgxl/
make && make install
cd contrib/
make && make install
```

编辑环境变量

```
# vim ~/.bashrc
export PGHOME=/home/postgres/pgxl
export PGUSER=postgres
export LD_LIBRARY_PATH=$PGHOME/lib:$LD_LIBRARY_PATH
export PATH=$PGHOME/bin:$PATH
```

运行生效：

```
source ~/.bashrc
```

配置群集

在 `gtm` 节点，运行 `pgxc_ctl` ,生成配置文件：

```
[postgres@gtm ~]$ pgxc_ctl
PGXC$ prepare config empty #生成一个空的配置文件/home/postgres/pgxc_ctl/pgxc_ctl.conf
PGXC$ exit

# 或者运行
pgxc_ctl prepare
```

修改配置文件（2个子节点）

```
$HOME/pgxc_ctl/pgxc_ctl.conf
```

```
#!/usr/bin/env bash
#
# Postgres-XC Configuration file for pgxc_ctl utility.
#
# Configuration file can be specified as -c option from pgxc_ctl command. Default is
# $PGXC_CTL_HOME/pgxc_ctl.org.
#
# This is bash script so you can make any addition for your convenience to configure
# your Postgres-XC cluster.
#
# Please understand that pgxc_ctl provides only a subset of configuration which
pgxc_ctl
# provide. Here's several several assumptions/restrictions pgxc_ctl depends on.
#
```

```

# 1) All the resources of pgxc nodes has to be owned by the same user. Same user
means
# user with the same user name. User ID may be different from server to server.
# This must be specified as a variable $pgxcOwner.
#
# 2) All the servers must be reachable via ssh without password. It is highly
recommended
# to setup key-based authentication among all the servers.
#
# 3) All the databases in coordinator/datanode has at least one same superuser.
Pgxc_ctl
# uses this user to connect to coordinators and datanodes. Again, no password
should
# be used to connect. You have many options to do this, pg_hba.conf, pg_ident.conf
and
# others. Pgxc_ctl provides a way to configure pg_hba.conf but not pg_ident.conf.
This
# will be implemented in the later releases.
#
# 4) Gtm master and slave can have different port to listen, while coordinator and
datanode
# slave should be assigned the same port number as master.
#
# 5) Port number of a coordinator slave must be the same as its master.
#
# 6) Master and slave are connected using synchronous replication. Asynchronous
replication
# have slight (almost none) chance to bring total cluster into inconsistent state.
# This chance is very low and may be negligible. Support of asynchronous
replication
# may be supported in the later release.
#
# 7) Each coordinator and datanode can have only one slave each. Cascaded replication
and
# multiple slave are not supported in the current pgxc_ctl.
#
# 8) Killing nodes may end up with IPC resource leak, such as semafor and shared
memory.
# Only listening port (socket) will be cleaned with clean command.
#
# 9) Backup and restore are not supported in pgxc_ctl at present. This is a big task
and
# may need considerable resource.
#
#=====
===
#
#
# pgxcInstallDir variable is needed if you invoke "deploy" command from pgxc_ctl
utility.
# If don't you don't need this variable.
pgxcInstallDir=$HOME/pgxc

pgxlDATA=$PGHOME/data
#---- OVERALL -----
-----
#
pgxcOwner=$USER # owner of the Postgres-XC databaseo cluster. Here, we use this

```

```

# both as linux user and database user. This must be
# the super user of each coordinator and datanode.
pgxcUser=$pgxcOwner      # OS user of Postgres-XC owner

tmpDir=$PGHOME/tmp        # temporary dir used in XC servers
localTmpDir=$tmpDir       # temporary dir used here locally

configBackup=n            # If you want config file backup, specify y to this
value.
configBackupHost=pgxc-linker # host to backup config file
configBackupDir=$HOME/pgxc  # Backup directory
configBackupFile=pgxc_ctl.bak # Backup file name --> Need to synchronize when
original changed.

dataDirRoot=$HOME/DATA/pgxl/nodes

#---- GTM -----
-----

# GTM is mandatory. You must have at least (and only) one GTM master in your
Postgres-XC cluster.
# If GTM crashes and you need to reconfigure it, you can do it by pgxc_update_gtm
command to update
# GTM master with others. Of course, we provide pgxc_remove_gtm command to remove
it. This command
# will not stop the current GTM. It is up to the operator.

#---- Overall -----
gtmName=(gtm)

#---- GTM Master -----

#---- Overall ----
gtmMasterServer=(gtm)
gtmMasterPort=6666
gtmMasterDir=$pgxlDATA/nodes/gtm

#---- Configuration ---
gtmExtraConfig=(none) # Will be added gtm.conf for both Master and Slave (done at
initilization only)
gtmMasterSpecificExtraConfig=(none) # Will be added to Master's gtm.conf (done at
initialization only)

#---- GTM Slave -----

# Because GTM is a key component to maintain database consistency, you may want to
configure GTM slave
# for backup.

#---- Overall -----
gtmSlave=y                # Specify y if you configure GTM Slave. Otherwise, GTM
slave will not be configured and
                           # all the following variables will be reset.

gtmSlaveName=(gtmSlave)
gtmSlaveServer=(gtm)      # value none means GTM slave is not available. Give none
if you don't configure GTM Slave.
gtmSlavePort=(20001)      # Not used if you don't configure GTM slave.
gtmSlaveDir=$pgxlDATA/nodes/gtmSlave # Not used if you don't configure GTM slave.

```

```

# Please note that when you have GTM failover, then there will be no slave available
until you configure the slave
# again. (pgxc_add_gtm_slave function will handle it)

#---- Configuration ----
gtmSlaveSpecificExtraConfig=(none) # Will be added to Slave's gtm.conf (done at
initialization only)

#---- GTM Proxy -----
-----

# GTM proxy will be selected based upon which server each component runs on.
# When fails over to the slave, the slave inherits its master's gtm proxy. It should
be
# reconfigured based upon the new location.
#
# To do so, slave should be restarted. So pg_ctl promote -> (edit postgresql.conf
and recovery.conf) -> pg_ctl restart
#
# You don't have to configure GTM Proxy if you don't configure GTM slave or you are
happy if every component connects
# to GTM Master directly. If you configure GTL slave, you must configure GTM proxy
too.

#---- Shortcuts -----
gtmProxyDir=$pgxlDATA/nodes/gtm_proxy

#---- Overall -----
gtmProxy=y          # Specify y if you configure at least one GTM proxy. You may
not configure gtm proxies
                    # only when you don't configure GTM slaves.
                    # If you specify this value not to y, the following parameters
will be set to default empty values.
                    # If we find there're no valid Proxy server names (means,
every servers are specified
                    # as none), then gtmProxy value will be set to "n" and all the
entries will be set to
                    # empty values.
gtmProxyNames=(gtm_pxy1) # No used if it is not configured
gtmProxyServers=(node1)  # Specify none if you don't configure it.
gtmProxyPorts=(6666)     # Not used if it is not configured.
gtmProxyDirs=($gtmProxyDir) # Not used if it is not configured.

#---- Configuration ----
gtmPxyExtraConfig=none  # Extra configuration parameter for gtm_proxy.
Coordinator section has an example.

#---- Coordinators -----
-----

#---- shortcuts -----
coordMasterDir=$dataDirRoot/coord_master
coordSlaveDir=$HOME/coord_slave
coordArchLogDir=$HOME/coord_archlog

#---- Overall -----
coordNames=(coord1)     # Master and slave use the same name
coordPorts=(5432)       # Master server listening ports
poolerPorts=(6667)      # Master pooler ports

```

```

coordPgHbaEntries=(0.0.0.0/0)    # Assumes that all the coordinator (master/slave)
accepts

                                # the same connection
                                # This entry allows only $pgxcOwner to
connect.

                                # If you'd like to setup another
connection, you should

                                # supply these entries through files
specified below.
#coordPgHbaEntries=(127.0.0.1/32) # Same as above but for IPv4 connections

#---- Master -----
coordMasterServers=(node1)       # none means this master is not available
coordMasterDirs=($coordMasterDir)
coordMaxWALsender=0 # max_wal_senders: needed to configure slave. If zero value is
specified,
                                # it is expected to supply this parameter explicitly by
external files
                                # specified in the following. If you don't configure slaves,
leave this value to zero.
coordMaxWALSenders=($coordMaxWALsender)
                                # max_wal_senders configuration for each coordinator.

#---- Slave -----
coordSlave=n                     # Specify y if you configure at least one coordinator slave.
Otherwise, the following
                                # configuration parameters will be set to empty values.
                                # If no effective server names are found (that is, every
servers are specified as none),
                                # then coordSlave value will be set to n and all the following
values will be set to
                                # empty values.
#coordSlaveSync=n               # Specify to connect with synchronized mode.
#coordSlaveServers=()           # none means this slave is not available
#coordSlavePorts=()             # coordinator slave listening ports
#coordSlavePoolerPorts=()       # coordinator slave pooler ports
#coordSlaveDirs=()
#coordArchLogDirs=()

#---- Configuration files---
# Need these when you'd like setup specific non-default configuration
# These files will go to corresponding files for the master.
# You may supply your bash script to setup extra config lines and extra pg_hba.conf
entries
# Or you may supply these files manually.
coordExtraConfig=coordExtraConfig # Extra configuration file for coordinators.
                                # This file will be added to all the coordinators'
                                # postgresql.conf
# Please note that the following sets up minimum parameters which you may want to
change.
# You can put your postgresql.conf lines here.
cat > $coordExtraConfig <<EOF
#=====
# Added to all the coordinator postgresql.conf
# Original: $coordExtraConfig
log_destination = 'stderr'
logging_collector = on
log_directory = 'pg_log'

```

```

listen_addresses = '*'
max_connections = 100
hot_standby = off
EOF

# Additional Configuration file for specific coordinator master.
# You can define each setting by similar means as above.
#coordSpecificExtraConfig=()
#coordSpecificExtraPgHba=()

#---- Datanodes -----
-----

#---- Shortcuts -----
datanodeMasterDir=$dataDirRoot/nodes/dn_master
datanodeSlaveDir=$dataDirRoot/dn_slave
datanodeArchLogDir=$dataDirRoot/datanode_archlog

#---- Overall -----
primaryDatanode=node1 # Primary Node.
datanodeNames=(node1)
datanodePorts=(5433)    # Master and slave use the same port!
datanodePoolerPorts=(6668) # Master and slave use the same port!
datanodePgHbaEntries=(0.0.0.0/0) # Assumes that all the coordinator (master/slave)
accepts
                                # the same connection
                                # This list sets up pg_hba.conf for $pgxcOwner
user.
                                # If you'd like to setup other entries, supply
them
                                # through extra configuration files specified
below.
#datanodePgHbaEntries=(127.0.0.1/32) # Same as above but for IPv4 connections

#---- Master -----
datanodeMasterServers=(node1) # none means this master is not available.
                                # This means that there should be
the master but is down.
                                # The cluster is not operational
until the master is
                                # recovered and ready to run.
datanodeMasterDirs=($datanodeMasterDir )
datanodeMaxWalSender=4         # max_wal_senders: needed to
configure slave. If zero value is
                                # specified, it is expected this
parameter is explicitly supplied
                                # by external configuration files.
                                # If you don't configure slaves,
leave this value zero.
datanodeMaxWALSenders=($datanodeMaxWalSender)
                                # max_wal_senders configuration for each datanode

#---- Slave -----
datanodeSlave=n                # Specify y if you configure at least one coordiantor slave.
Otherwise, the following
                                # configuration parameters will be set to empty values.
                                # If no effective server names are found (that is, every
servers are specified as none),

```



```

        # then datanodeSlave value will be set to n and all the
following values will be set to
        # empty values.
datanodeSlaveServers=(node1)    # value none means this slave is not available
datanodeSlavePorts=(15433)    # Master and slave use the same port!
datanodeSlavePoolerPorts=(20012)    # Master and slave use the same port!
datanodeSlaveSync=y    # If datanode slave is connected in synchronized mode
datanodeSlaveDirs=($datanodeSlaveDir )
datanodeArchLogDirs=($datanodeArchLogDir )

# ---- Configuration files ----
# You may supply your bash script to setup extra config lines and extra pg_hba.conf
entries here.
# These files will go to corresponding files for the master.
# Or you may supply these files manually.
datanodeExtraConfig=datanodeExtraConfig
cat > $datanodeExtraConfig <<EOF
#=====
# Added to all the datanode postgresql.conf
# Original: $datanodeExtraConfig
log_destination = 'stderr'
logging_collector = on
log_directory = 'pg_log'
listen_addresses = '*'
max_connections = 100
hot_standby = off
EOF
# Additional Configuration file for specific datanode master.
# You can define each setting by similar means as above.
datanodeSpecificExtraConfig=(none)
datanodeSpecificExtraPgHba=(none)

```

- 备份配置到 **node1**

```

scp /home/postgres/pgxc_ctl/pgxc_ctl.conf
postgres@node1:/home/postgres/pgxc_ctl/pgxc_ctl.conf

```

初始化所有配置:

```

#初始化所有节点配置
[postgres@gtm ~]$ pgxc_ctl -c /home/postgres/pgxc_ctl/pgxc_ctl.conf init all

# 查看节点状态
[postgres@docker-hbase root]$ pgxc_ctl
/usr/bin/bash
Installing pgxc_ctl_bash script as /home/postgres/pgxc_ctl/pgxc_ctl_bash.
Installing pgxc_ctl_bash script as /home/postgres/pgxc_ctl/pgxc_ctl_bash.
Reading configuration using /home/postgres/pgxc_ctl/pgxc_ctl_bash --home
/home/postgres/pgxc_ctl --configuration /home/postgres/pgxc_ctl/pgxc_ctl.conf
Finished reading configuration.
***** PGXC_CTL START *****

Current directory: /home/postgres/pgxc_ctl
PGXC monitor all
Running: gtm master
Running: gtm slave
Running: gtm proxy gtm_pxy1
Running: coordinator master coord1

```

```
Running: datanode master node1
```

启动、关闭集群

```
pgxc_ctl -c /home/postgres/pgxc_ctl/pgxc_ctl.conf start all
```

```
pgxc_ctl -c /home/postgres/pgxc_ctl/pgxc_ctl.conf stop all
```

查看运行状态：

```
# 在node1上连接coord1(coordinator)节点
```

```
psql -p 5432 postgres
```

```
psql (PGXL 10alpha2, based on PG 10beta3 (Postgres-XL 10alpha2))
```

```
Type "help" for help.
```

```
postgres=# select * from pgxc_node;
```

```
 node_name | node_type | node_port | node_host | nodeis_primary | nodeis_preferred |
 node_id
```

```
-----+-----+-----+-----+-----+-----+-----+
-----
```

```
 coord1    | C          |      5432 | node1     | f              | f              |
1885696643
```

```
 node1     | D          |      5433 | node1     | t              | t              |
1148549230
```

```
(2 rows)
```

```
postgres=#
```

```
# 在node1上连接datanode节点
```

```
psql -p 5433 postgres
```

```
psql (PGXL 10alpha2, based on PG 10beta3 (Postgres-XL 10alpha2))
```

```
Type "help" for help.
```

```
postgres=# select * from pgxc_node;
```

```
 node_name | node_type | node_port | node_host | nodeis_primary | nodeis_preferred |
 node_id
```

```
-----+-----+-----+-----+-----+-----+
-----
```

```
 coord1    | C          |      5432 | node1     | f              | f              |
1885696643
```

```
 node1     | D          |      5433 | node1     | t              | t              |
1148549230
```

```
(2 rows)
```

- 也可以在 GTM 上进行操作

```
# psql -h host -p port dbname
```

```
# 连接到node1节点的postgres库,默认为postgres库
```

```
psql -h node1 -p 5432 postgres
```

```
# 查看数据库
```

```
\l或者\l+
```

```
select * from pg_database;
```

```
# 切换数据库
```

```
\c test;
```

```
# 查看表
```

```
\dt或者  
select * from pg_tables where schemaname='public';
```

#查看表结构

```
\d table;
```