**数据库基础环境部署**

**操作手册**

**（rhel 7.4 PostgresQL 9.6.2）**

**目录**

[1 操作系统及软件版本 3](#_Toc519771097)

[2 PostgresQL安装流程 3](#_Toc519771098)

[2.1 下载安装介质 3](#_Toc519771099)

[2.2 新建用户及创建安装路径 3](#_Toc519771100)

[2.3 安装缺失软件包 3](#_Toc519771101)

[2.4 编译安装，并安装第三方组织的PostgresQL工具代码 4](#_Toc519771102)

[2.5 设置环境变量 4](#_Toc519771103)

[2.6 检验环境变量是否配置正确 5](#_Toc519771104)

[2.7 初始化数据库并进行监听配置 5](#_Toc519771105)

[3 数据库相关操作 6](#_Toc519771106)

[3.1 数据库启停 6](#_Toc519771107)

[3.2 创建数据库及数据库用户 6](#_Toc519771108)

[3.3 查看监听参数 6](#_Toc519771109)

[4 PostgresQL主从流复制 7](#_Toc519771110)

[4.1 主库配置 7](#_Toc519771111)

[4.2 从库配置 8](#_Toc519771112)

[4.3 查看复制状态 8](#_Toc519771113)

# 操作系统及软件版本

操作系统为：数据中心redhat 7.4数据库专用操作系统模板。

PostgresQL版本：PostgresQL-9.6.2。

# PostgresQL安装流程

## 下载安装介质

下载安装包postgresql-9.6.2.tar.gz待用。

## 新建用户及创建安装路径

使用root用户执行：

groupadd -g 6001 postgres

useradd postgres -u 6001 -g 6001 #创建新用户组postgres，在其中创建新用户postgres

passwd postgres

输入：Clic\_1234 #修改用户postgres密码

mkdir -p /postgresql #创建安装目录文件夹posgresql

mkdir -p /data #创建数据存储文件夹data

chown -R postgres:postgres /postgresql #postgresql 是pg程序的安装目录

chown -R postgres:postgres /data #存放数据，创建文件系统的过程省略，磁盘用ext3的格式

之后，将用postgres用户控制PostgresQL数据库的相关操作。

## 安装缺失软件包

在CHINALIFE RHEL 7.4中，一些安装过程中所需的软件包缺失，需手动安装：

下载rpm包：

ncurses-devel-5.7-4.20090207.el6.x86\_64.rpm；

readline-devel-6.0-4.el6.x86\_64.rpm；

zlib-devel-1.2.3-29.el6.x86\_64.rpm（版本可更新）

使用root用户在同文件夹下执行：

rpm -ivh ncurses-devel-5.7-4.20090207.el6.x86\_64.rpm

rpm -ivh readline-devel-6.0-4.el6.x86\_64.rpm

rpm -ivh zlib-devel-1.2.3-29.el6.x86\_64.rpm

系统如若已安装yum，也可直接使用yum install命令安装以上三个包。

## 编译安装，并安装第三方组织的PostgresQL工具代码

使用root用户完成下列操作：

#将之前下载的PostgresQL安装包移动到之前建立的/postgresql文件夹中

mv /路径/postgresql-9.6.2.tar.gz /postgresql

cd /postgresql #进入到postgresql文件夹中

tar zxvf postgresql-9.6.2.tar.gz #解压

cd postgresql-9.6.2

./configure --prefix=/postgresql/config #执行configure，为软件指定安装目录，并检查软件包

make

make install #执行编译

#安装contrib目录下的一些工具，是第三方组织的一些工具代码，建议安装

cd /postgresql/postgresql-9.6.2/contrib #这个是软件程序安装包所在的地方

make

make install

## 设置环境变量

切换到postgres用户，完成以下操作：

cd /home/postgres #确保已经处于用户的根目录下

vi .bash\_profile #修改配置文件

在配置文件中export PATH语句之前添加如下内容：

export PGHOME=/postgresql/config

export PGDATA=/data/postgresql

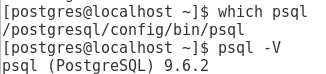
PATH=$PATH:$HOME/.local/bin:$HOME/bin:$PGHOME/bin

添加完成之后在vi命令行模式下输入:wq保存并退出。

## 检验环境变量是否配置正确

使用postgres用户：

退出登录之后重新登录为postgres用户，切换任意目录，输入which psql以及psql –V可以分别查看psql客户端路径以及postgresql的版本。如下：



此时说明环境变量配置正确。

## 初始化数据库并进行监听配置

使用postgres用户执行：

initdb -D /data/postgresql #初始化数据库

cd /data/postgresql

vi postgresql.conf #进行监听的配置

在postgresql.conf中，修改下列变量并取消注释：

listen\_addresses = '\*' #允许监听所有来的地址。如果不带开，远程连接会被拒绝。

logging\_collector = on #把logging打开

log\_line\_prefix = '%t' #把logging的时间戳打印出来

下面两个默认就是配好的：

timezone='PRC' #把时区改为中国时区

log\_timezone='PRC'

添加完成之后在vi命令行模式下输入:wq保存并退出。

使用postgres用户在/data/postgresql文件夹下执行：

vi pg\_hba.conf

修改pg\_hba.conf文件。如果这里缺少配置，客户端连接会报错。全开放的话就是0.0.0.0/0

host all all 10.20.0.0/16 trust

host all all 10.21.0.0/16 trust

# 数据库相关操作

## 数据库启停

使用postgres用户执行：

pg\_ctl -D /data/postgresql -l logfile start #启动数据库

pg\_ctl stop #停止数据库

如后续需要更新数据库配置，使用：

pg\_ctl reload（仅在server启动后生效） #用于配置文件修改后数据库更新

## 创建数据库及数据库用户

使用postgres用户执行：

psql #进入PostgresQL前端

在PostgresQL前端psql（postgres=#）下：

CREATE USER cmds WITH PASSWORD 'q1w2e3r4'; #创建用户和密码

CREATE DATABASE cmds OWNER cmds; #创建数据库，并指定数据库的用户

GRANT ALL PRIVILEGES ON DATABASE cmds TO cmds; #将数据库的所有权限授权给架构用户

根据需要选择进行PostgresQL数据库的相关操作，以下是几个具体范例：

CREATE tablespace tablespace01 location '/data/cmds\_data'; #新创建表空间，不能放在postgres data的目录里面，可以在上一层

DROP tablespace IF EXISTS tablespace01; #删除表空间

ALTER database cmds SET tablespace tablespace01; #调整数据库的默认表空间

## 查看监听参数

使用postgres用户执行：

cd /data/postgresql #回到该文件夹，修改配置文件

vi postgresql.conf

修改配置文件postgresql.conf，并且重启PostgreSQL方能生效（停止当前PostgresQL数据库服务之后再修改）

添加如下字段：

#------------------------------------------------------------------------------

# PG\_STAT\_STATEMENTS OPTIONS

#------------------------------------------------------------------------------

shared\_preload\_libraries = 'pg\_stat\_statements'

pg\_stat\_statements.max = 1000

pg\_stat\_statements.track = all

修改配置完成后，启动数据库服务，进入psql：

CREATE extension pg\_stat\_statements; #首次查看时需要

（可选）\x; #更改显示格式

SELECT query, calls, total\_time, (total\_time/calls) as average, rows, 100.0 \* shared\_blks\_hit /nullif(shared\_blks\_hit + shared\_blks\_read, 0) AS hit\_percent FROM pg\_stat\_statements ORDER BY average DESC LIMIT 10; #选择字段显示结果

# PostgresQL主从流复制

## 主库配置

使用postgres用户进入/data/postgresql修改pg\_hba.conf文件：

执行：

vi pg\_hba.conf #编辑pg\_hba.conf文件

进入编辑界面后，增添如下语句：

host replication replica 20.18.208.5/32 trust #新增用户，此处20.18.208.5为从库IP地址，实际操作中按具体情况修改

退出编辑界面，启动数据库：

pg\_ctl -D /data/postgresql -l logfile start #启动数据库

进入psql，为replica用户设置密码及备份权限等：

postgres# CREATE ROLE replica login replication encrypted password 'replica'; #设置备份权限并设定密码

退出psql，在/data/postgresql文件夹下修改postgresql.conf中以下变量并取消注释：

wal\_level = hot\_standby #设置主从流

max\_wal\_senders = 32 #设置复制连接的流的最大数目，决定从数据库最大数目

wal\_keep\_segments = 256 #设置流复制保留的最多的xlog数目

wal\_sender\_timeout = 60s #设置流复制主机发送数据的超时时间

max\_connections = 100 #该设置注意，从库的max\_connections必须要大于主库的max\_connections

修改完成后，重启主数据库，以使配置生效：

pg\_ctl restart #重启数据库

## 从库配置

在从数据库所在主机上：

备份文件夹地址为/data/postgresql/data2。

使用postgres用户执行：

pg\_basebackup -F p --progress -D /data/pgsql/data2 -h 20.18.208.29 -p 5432 -U replica --password #备份主机数据库并传送到设定路径，此处20.18.208.29为主库IP地址，实际操作中按具体情况修改

输入上面设置的密码replica以完成备份。

进入/data/postgresql/data2文件夹：

cd /data/postgresql/data2

cp /postgresql/config/share/recovery.conf.sample /data/postgresql/data2 #从安装文件夹处复制recovery.conf到从库文件夹

vi recovery.conf #修改配置

进入编辑界面，修改如下变量并取消注释：

standby\_mode = on #设定从库

primary\_conninfo = 'host=20.18.208.29 port=5432 user=replica password=replica' #说明这台机器对应主库的IP地址，端口以及从库的用户名与密码

recovery\_target\_timeline = 'latest' #说明复制同步最新数据

完成编辑后退出编辑界面，启动从数据库：

pg\_ctl -D /data/postgresql/data2 -l logfile start #启动数据库

## 查看复制状态

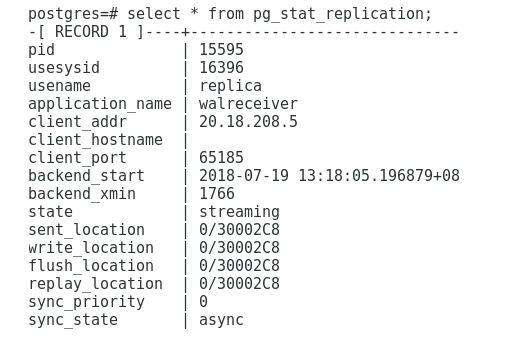
通过前两步的配置完成了主从配置，接下来在主数据库上查看从库的复制状态：

在主库所在主机上进入psql：

postgres=# \x; #更改显示格式

postgres=# SELECT \* FROM pg\_stat\_replication;

输出该主库对应从库的记录，如下图：



可以通过此命令查看从库的复制状态。