迁移步骤

1. **系统信息**
2. 服务器列表

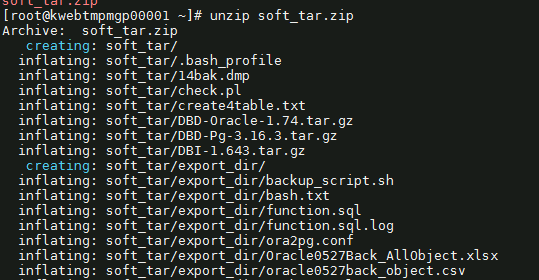
|  |  |  |  |
| --- | --- | --- | --- |
| 机器 | IP | OS系统 | 实例规格 |
| Postgres14.8 | 10.28.160.242 | CentOS release 6.10(Final) | 88U755G |

1. **系统环境安装准备**
2. 安装pg14.8
   1. 上传文件



* 1. 解压文件

unzip soft\_tar.zip

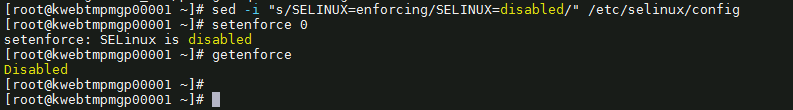


* 1. 禁用SELinux强制模式

sed -i "s/SELINUX=enforcing/SELINUX=disabled/" /etc/selinux/config

setenforce 0

getenforce



* 1. 关闭防火墙

service iptables stop

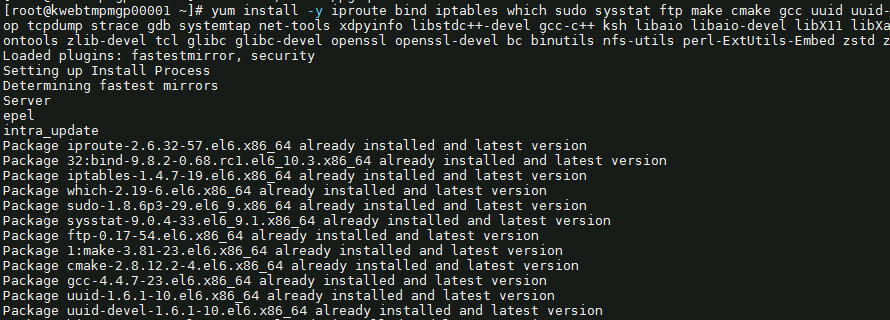
* 1. 设置变量

PGDATA=’/data01/pgsql’



* 1. 安装依赖包

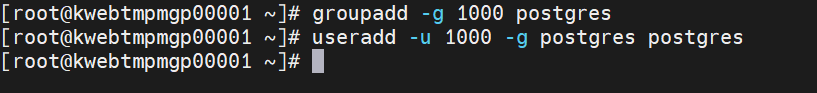
yum install -y iproute bind iptables which sudo sysstat ftp make cmake gcc uuid uuid-devel bison flex perl perl-devel python-devel readline readline-devel libxml2 libxml2-devel iotop tcpdump strace gdb systemtap net-tools xdpyinfo libstdc++-devel gcc-c++ ksh libaio libaio-devel libX11 libXau libXi libXtst libXrender libXrender-devel libgcc libstdc++ libstdc++-devel libxcb make smartmontools zlib-devel tcl glibc glibc-devel openssl openssl-devel bc binutils nfs-utils perl-ExtUtils-Embed zstd zstd-devel libcurl libcurl-devel lz4 lz4-devel libicu libicu-devel autoconf



* 1. 创建用户和组

groupadd -g 1000 postgres

useradd -u 1000 -g postgres postgres



* 1. 创建数据库目录

mkdir -p $PGDATA



* 1. 需要先上传到root

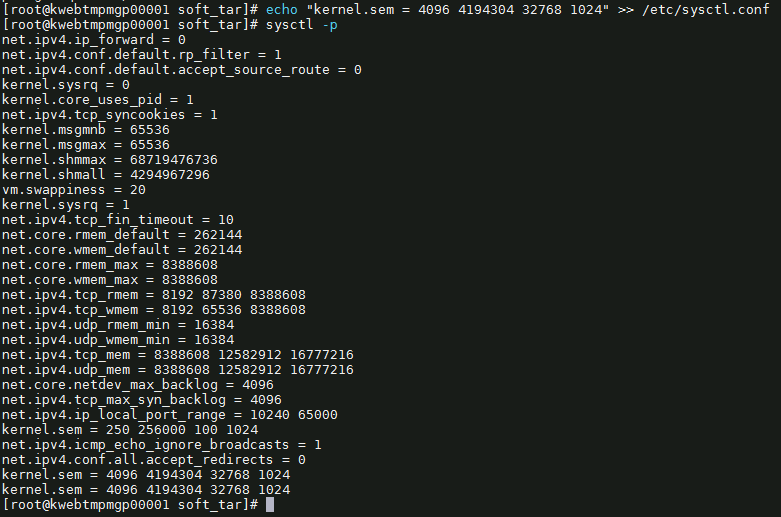
cd soft\_tar

tar -zxvf postgresql-14.8.tar.gz -C /data01/app/install 

* 1. 修改服务器内存和信号量

echo "kernel.sem = 4096 4194304 32768 1024" >> /etc/sysctl.conf

sysctl -p



* 1. 修改服务器资源限制

echo "postgres soft nproc unlimited" >> /etc/security/limits.conf

echo "postgres hard nproc unlimited" >> /etc/security/limits.conf

echo "postgres soft nofile 1024000" >> /etc/security/limits.conf

echo "postgres hard nofile 1024000" >> /etc/security/limits.conf

echo "postgres soft stack unlimited" >> /etc/security/limits.conf

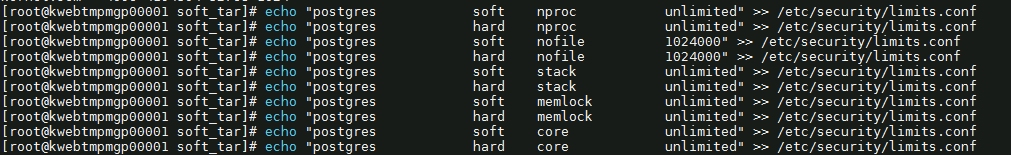
echo "postgres hard stack unlimited" >> /etc/security/limits.conf

echo "postgres soft memlock unlimited" >> /etc/security/limits.conf

echo "postgres hard memlock unlimited" >> /etc/security/limits.conf

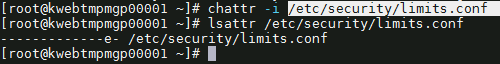
echo "postgres soft core unlimited" >> /etc/security/limits.conf

echo "postgres hard core unlimited" >> /etc/security/limits.conf



报错没有权限写入

chattr -i /etc/security/limits.conf

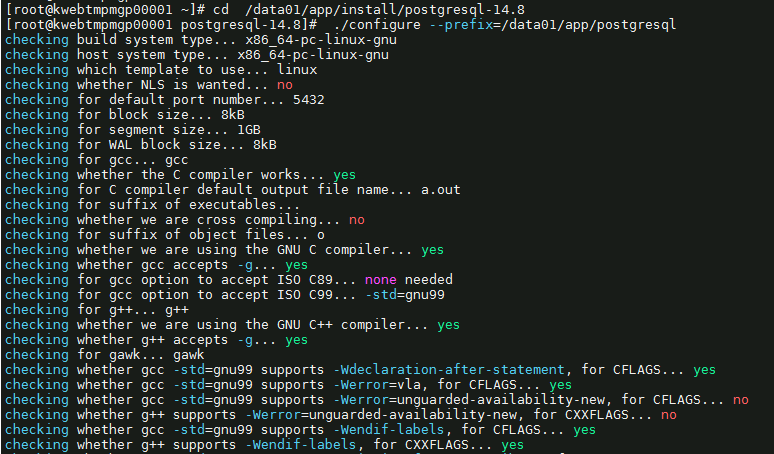


* 1. 编译包

cd /data01/app/install/postgresql-14.8

./configure --prefix=/data01/app/postgresql

make world && make install-world



* 1. 添加配置文件

su - postgres -c "echo 'export PG\_HOME=/data01/app/postgresql' >> /home/postgres/.bash\_profile"

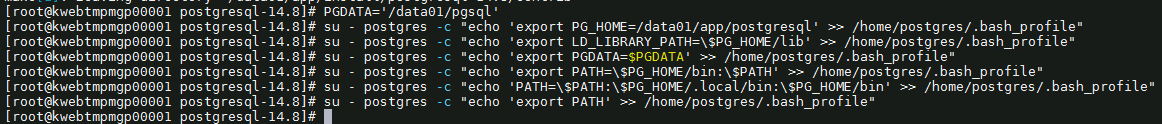
su - postgres -c "echo 'export LD\_LIBRARY\_PATH=\$PG\_HOME/lib' >> /home/postgres/.bash\_profile"

su - postgres -c "echo 'export PGDATA=$PGDATA' >> /home/postgres/.bash\_profile"

su - postgres -c "echo 'export PATH=\$PG\_HOME/bin:\$PATH' >> /home/postgres/.bash\_profile"

su - postgres -c "echo 'PATH=\$PATH:\$PG\_HOME/.local/bin:\$PG\_HOME/bin' >> /home/postgres/.bash\_profile"

su - postgres -c "echo 'export PATH' >> /home/postgres/.bash\_profile"



* 1. 初始化数据库

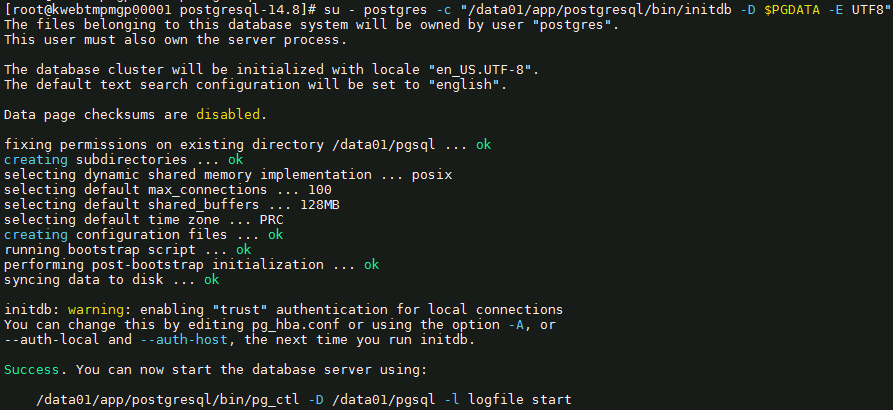
chown -R postgres:postgres /data01/app/install

chown -R postgres:postgres /data01/app/postgresql

chown -R postgres:postgres $PGDATA



su - postgres -c "/data01/app/postgresql/bin/initdb -D $PGDATA -E UTF8"



* 1. 创建归档日志目录

su - postgres -c "mkdir -p $PGDATA/archivedir"



* 1. 修改pg\_hba.conf

su - postgres -c "echo 'host all all 0/0 md5' >> $PGDATA/pg\_hba.conf"

su - postgres -c "echo 'host replication replica 0/0 md5' >> $PGDATA/pg\_hba.conf"



* 1. 配置postgresql.conf

if [ $(lscpu |grep '^CPU(s): ' | awk -F " " '{print $1}') == 'CPU(s):' ]

then

CPU=$(lscpu |grep '^CPU(s): ' | awk -F " " '{print $2}')

elif [ $(lscpu |grep '^CPU: ' | awk -F " " '{print $1}') == 'CPU:' ]

then

CPU=$(lscpu |grep '^CPU: ' | awk -F " " '{print $2}')

else

echo "没有符合的条件"

fi

MEM\_S=$(free -m|grep '^Mem:' | awk -F " " '{print expr $2/1024\*0.4}' | cut -d '.' -f1)GB

MEM\_E=$(free -m|grep '^Mem:' | awk -F " " '{print $2/1024\*0.5}' | cut -d '.' -f1)GB

sed -i "s/#listen\_addresses = 'localhost'/listen\_addresses = '\*'/" $PGDATA/postgresql.conf

sed -i "s/#port = 5432/port = 5432/" $PGDATA/postgresql.conf

sed -i "s/max\_connections = 100/max\_connections = 3000/" $PGDATA/postgresql.conf

# 开启大页，这个项也要开启

# sed -i "s/#huge\_pages = try/huge\_pages = on/" $PGDATA/postgresql.conf

sed -i "s/shared\_buffers = 128MB/shared\_buffers = $MEM\_S/" $PGDATA/postgresql.conf

sed -i "s/#work\_mem = 4MB/work\_mem = 16MB/" $PGDATA/postgresql.conf

sed -i "s/#wal\_buffers = -1/wal\_buffers = 16MB/" $PGDATA/postgresql.conf

sed -i "s/#checkpoint\_completion\_target = 0.9/checkpoint\_completion\_target = 0.9/" $PGDATA/postgresql.conf

sed -i "s/max\_wal\_size = 1GB/max\_wal\_size = 8GB/" $PGDATA/postgresql.conf

sed -i "s/min\_wal\_size = 80MB/min\_wal\_size = 2GB/" $PGDATA/postgresql.conf

sed -i "s/#archive\_mode = off/archive\_mode = on/" $PGDATA/postgresql.conf

sed -i "s/#archive\_command = ''/archive\_command = 'test ! -f \/data01\/pgsql\/archivedir\/%f \&\& cp %p \/data01\/pgsql\/archivedir\/%f'/" $PGDATA/postgresql.conf

sed -i "s/#default\_statistics\_target = 100/default\_statistics\_target = 100/" $PGDATA/postgresql.conf

sed -i "s/#log\_destination = 'stderr'/log\_destination = 'csvlog'/" $PGDATA/postgresql.conf

sed -i "s/#logging\_collector = off/logging\_collector = on/" $PGDATA/postgresql.conf

sed -i "s/#effective\_cache\_size = 4GB/effective\_cache\_size = $MEM\_E/" $PGDATA/postgresql.conf

sed -i "s/#random\_page\_cost = 4.0/random\_page\_cost = 1.1/" $PGDATA/postgresql.conf

sed -i "s/#maintenance\_io\_concurrency = 10/maintenance\_io\_concurrency = 200/" $PGDATA/postgresql.conf

sed -i "s/#max\_worker\_processes = 8/max\_worker\_processes = $CPU/" $PGDATA/postgresql.conf

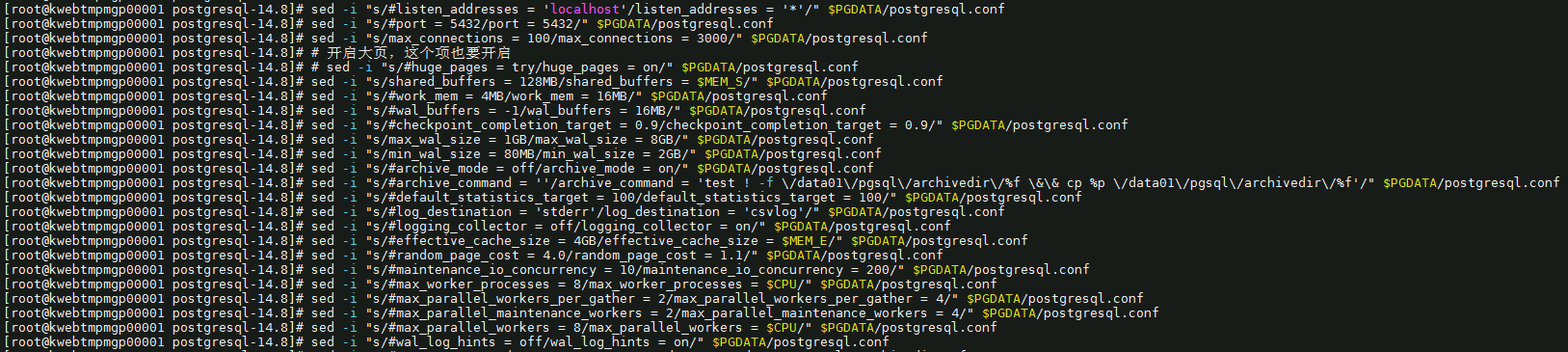
ccccsed -i "s/#max\_parallel\_workers\_per\_gather = 2/max\_parallel\_workers\_per\_gather = 4/" $PGDATA/postgresql.conf

sed -i "s/#max\_parallel\_maintenance\_workers = 2/max\_parallel\_maintenance\_workers = 4/" $PGDATA/postgresql.conf

sed -i "s/#max\_parallel\_workers = 8/max\_parallel\_workers = $CPU/" $PGDATA/postgresql.conf

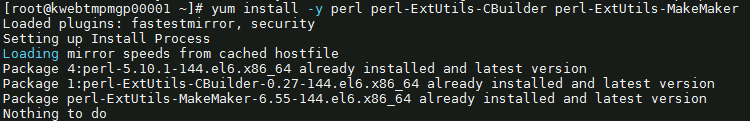
sed -i "s/#wal\_log\_hints = off/wal\_log\_hints = on/" $PGDATA/postgresql.conf

sed -i "s/#restore\_command = ''/restore\_command = 'cp \/data01\/pgsql\/archivedir\/%f %p'/" $PGDATA/ postgresql.conf

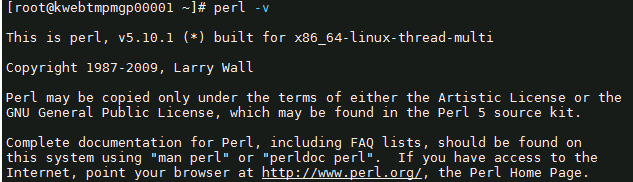


1. 安装ora2pg
   * 1. 安装perl

yum install -y perl perl-ExtUtils-CBuilder perl-ExtUtils-MakeMaker



perl -v



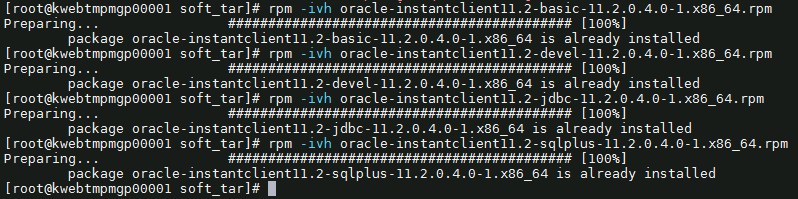
* + 1. 安装oracle客户端

rpm -ivh oracle-instantclient11.2-basic-11.2.0.4.0-1.x86\_64.rpm

rpm -ivh oracle-instantclient11.2-devel-11.2.0.4.0-1.x86\_64.rpm

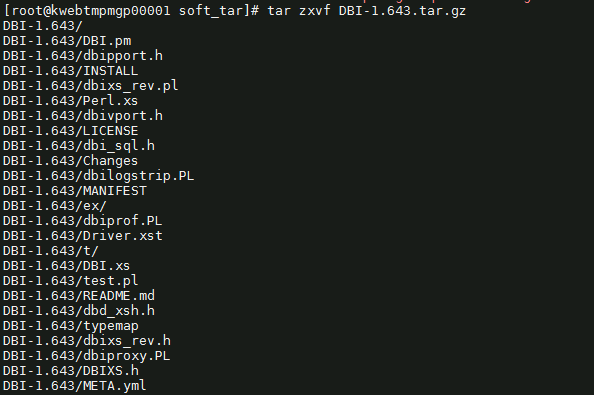
rpm -ivh oracle-instantclient11.2-jdbc-11.2.0.4.0-1.x86\_64.rpm

rpm -ivh oracle-instantclient11.2-sqlplus-11.2.0.4.0-1.x86\_64.rpm



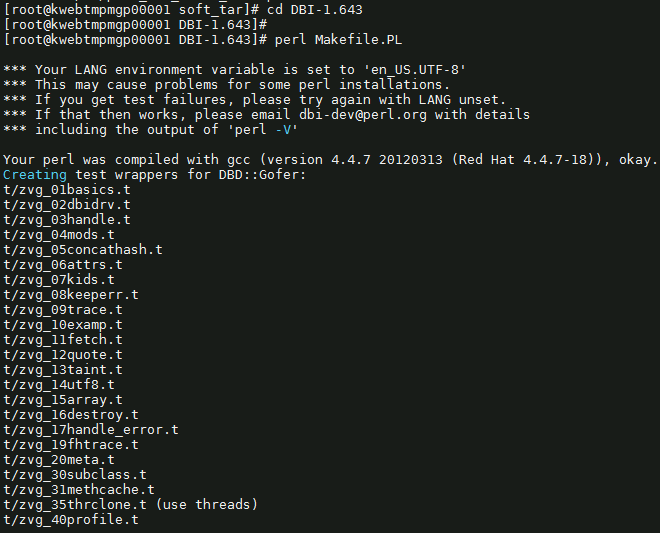
* + 1. 安装DBI模块

tar zxvf DBI-1.643.tar.gz

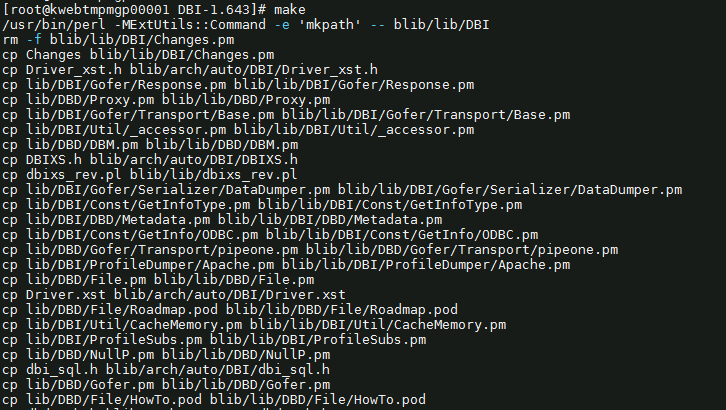


cd DBI-1.643

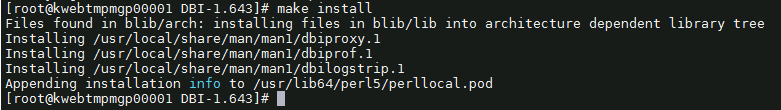
perl Makefile.PL



make



make install

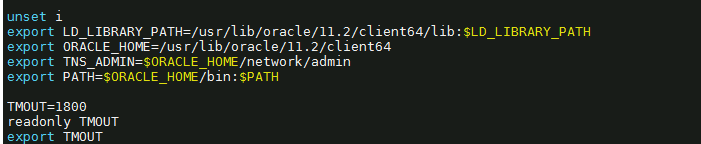


* + 1. 安装DBD::oracle驱动模块

vi /etc/profile

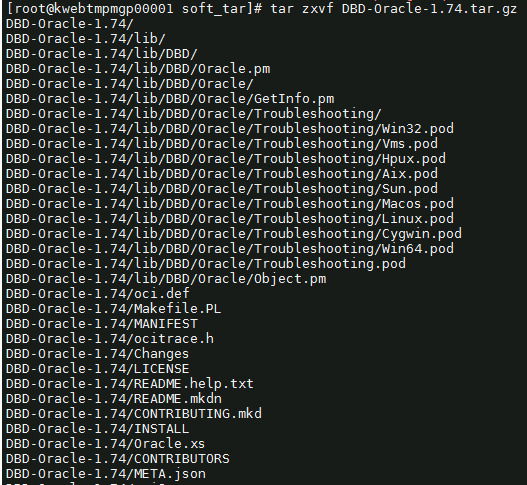
export LD\_LIBRARY\_PATH=/usr/lib/oracle/11.2/client64/lib

export ORACLE\_HOME=/usr/lib/oracle/11.2/client64



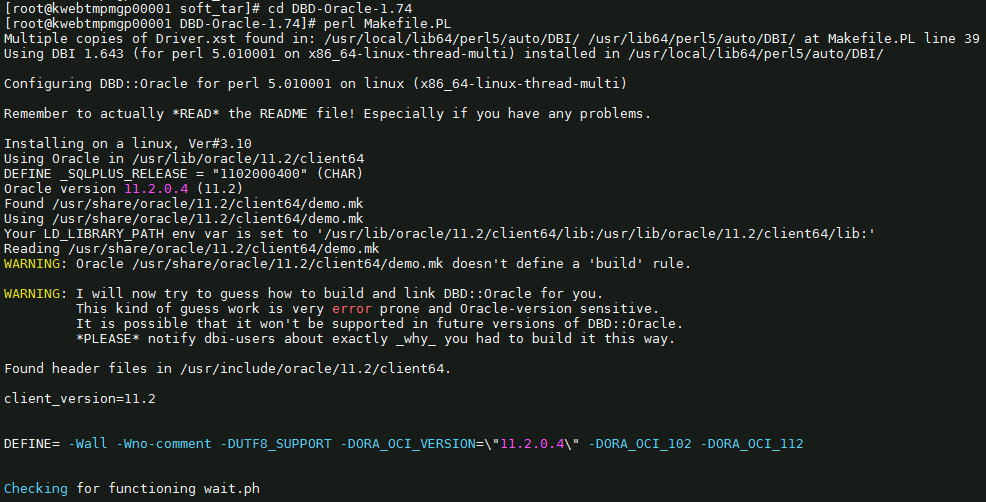
source /etc/profile

tar zxvf DBD-Oracle-1.74.tar.gz

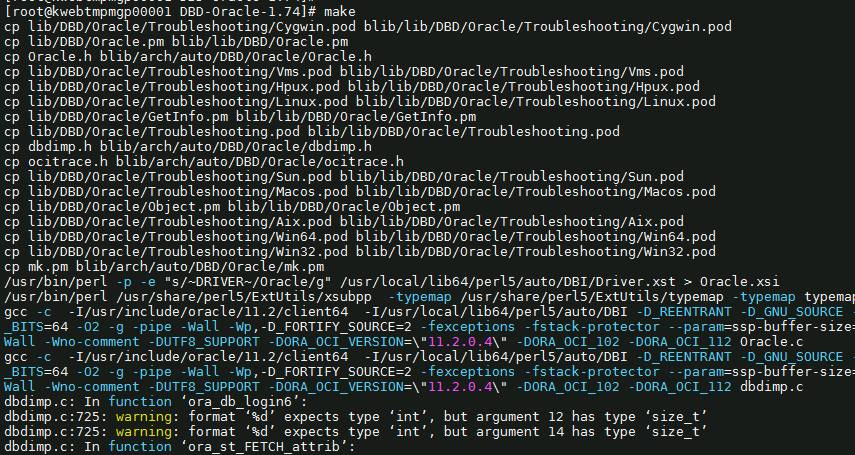


cd DBD-Oracle-1.74

perl Makefile.PL



make



make install

* + 1. 安装DBD:pg驱动模块

Vi /etc/profile

export POSTGRES\_INCLUDE="/data01/app/postgresql/include"

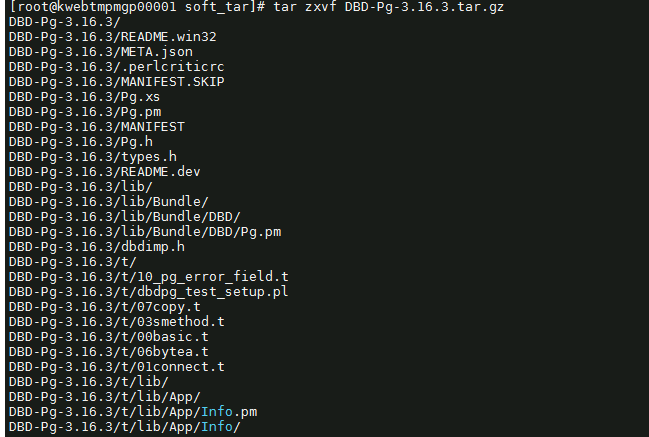
export POSTGRES\_LIB="/data01/app/postgresql/lib"

export POSTGRES\_HOME="/data01/app/postgresql"



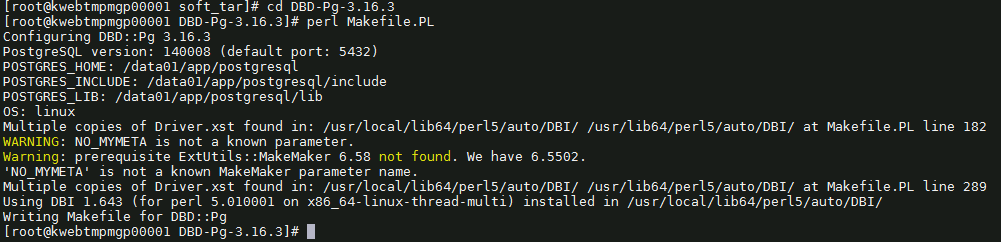
source /etc/profile

tar zxvf DBD-Pg-3.16.3.tar.gz

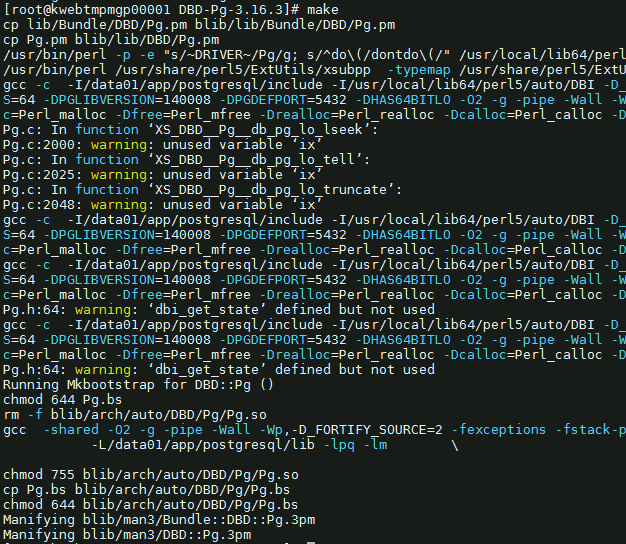


cd DBD-Pg-3.16.3

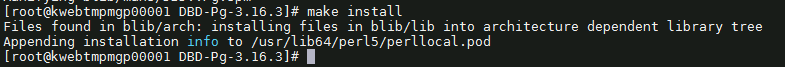
perl Makefile.PL



make

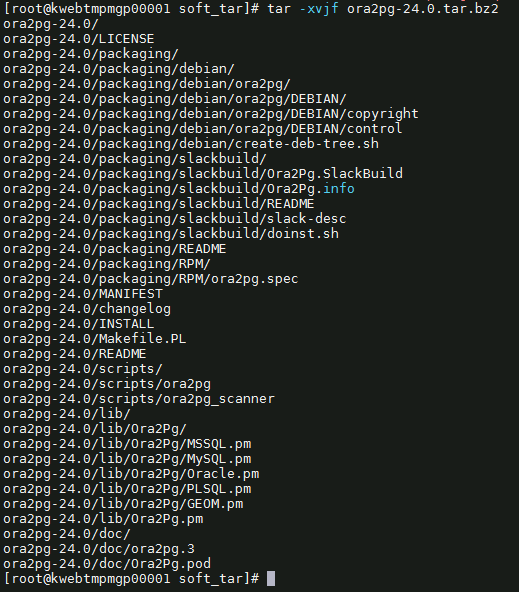


make install



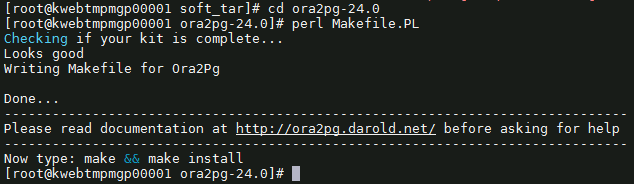
* + 1. 安装ora2pg

tar -xvjf ora2pg-24.0.tar.bz2

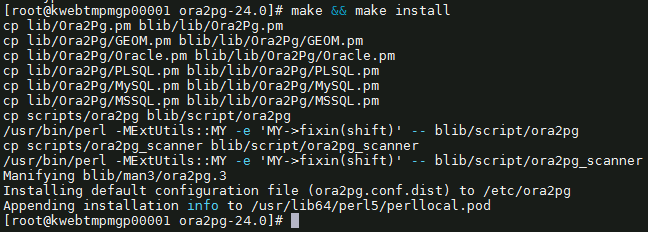


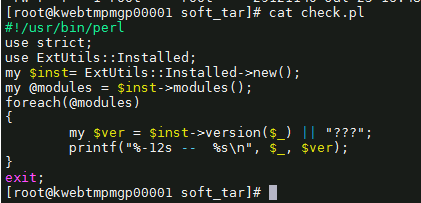
cd ora2pg-24.0

perl Makefile.PL

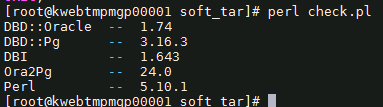


make && make install





perl check.pl



ora2pg



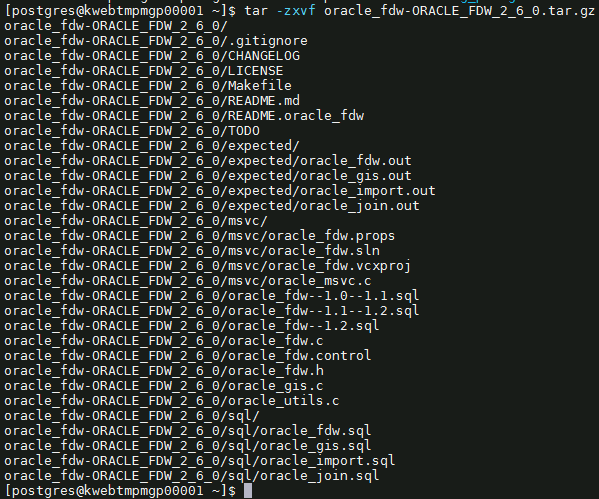
1. 安装oracle\_fdw
   1. 下载地址：

https://github.com/laurenz/oracle\_fdw

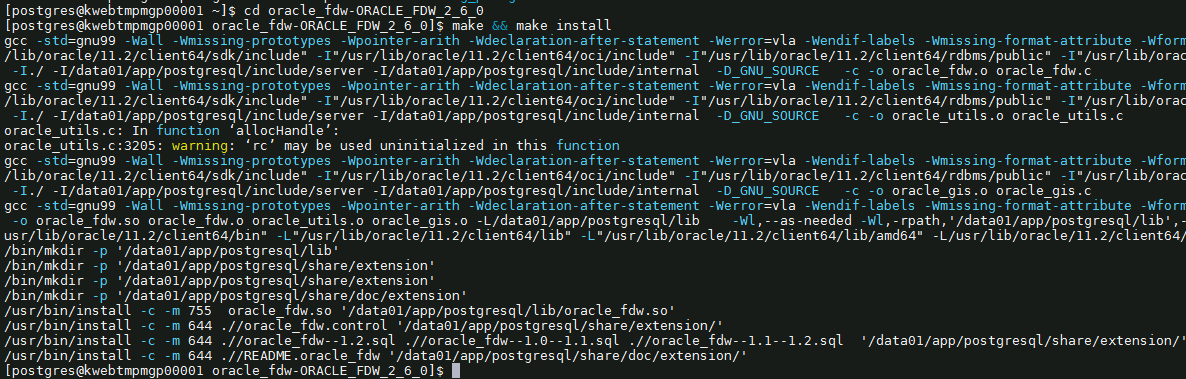
下载到 /home/postgres

* 1. 安装步骤

tar -zxvf oracle\_fdw-ORACLE\_FDW\_2\_6\_0.tar.gz



cd oracle\_fdw-ORACLE\_FDW\_2\_6\_0



* 1. 添加环境变量

vi .bash\_profile

PATH=$PATH:$HOME/bin

export PATH

export PG\_HOME=/data01/app/postgresql

export ORACLE\_HOME=/usr/lib/oracle/11.2/client64

export LD\_LIBRARY\_PATH=$PG\_HOME/lib

export PGDATA=/data01/pgsql

export PATH=$PG\_HOME/bin:$ORACLE\_HOME/bin:$PATH

PATH=$PATH:$PG\_HOME/.local/bin:$PG\_HOME/bin

export LD\_LIBRARY\_PATH=/usr/lib/oracle/11.2/client64/lib:$LD\_LIBRARY\_PATH

export TNS\_ADMIN=$ORACLE\_HOME/network/admin

export PATH

export PG\_HOME=/data01/app/postgresql

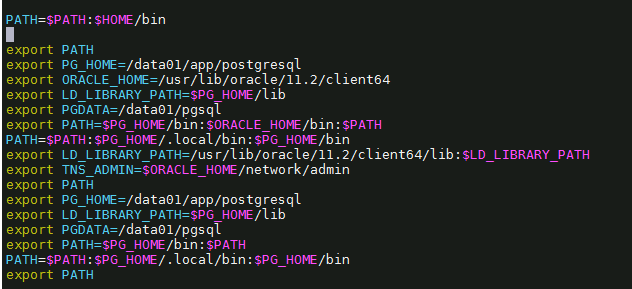
export LD\_LIBRARY\_PATH=$PG\_HOME/lib

export PGDATA=/data01/pgsql

export PATH=$PG\_HOME/bin:$PATH

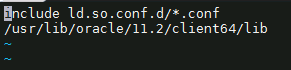
PATH=$PATH:$PG\_HOME/.local/bin:$PG\_HOME/bin

export PATH



* 1. 修改/etc/ld.so.conf添加一行

/usr/lib/oracle/11.2/client64/lib



运行ldconfig



* 1. 数据库加载插件



* 1. 新建tnsnames.ora

cd /usr/lib/oracle/11.2/client64/network/admin

vi tnsnames.ora

test\_oracle =

(DESCRIPTION =

(ADDRESS\_LIST=

(ADDRESS=(PROTOCOL=TCP)(HOST=MPMUAToracle01.beta.hic.cloud)(PORT=1521))

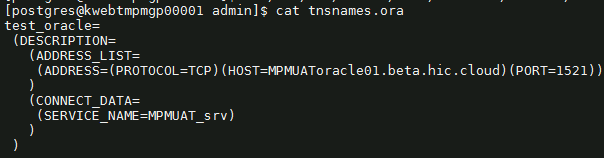
)

(CONNECT\_DATA=

(SERVICE\_NAME=MPMUAT\_srv)

)

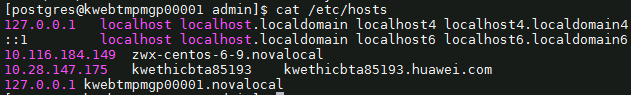
)



* 1. 修改/etc/hosts文件

127.0.0.1 kwebtmpmgp0001.novalocal

cat /etc/hosts



1. **数据库准备**
2. 查看原数据库信息
3. 数据库准备

2.1.新建用户

2.2新建表空间

2.3新建库

1. **迁移为了方便创建的函数**
2. 新建函数及表
3. 删除函数及表
4. **Oracle数据库表、索引迁移到PG库**
5. 使用flashsync导表结构
6. flashsync导表数据
7. **oracle数据库迁移约束、序列、视图、过程、函数、包到PG**
8. 约束
9. 序列
10. 视图
11. 过程
12. 函数
13. 包
14. **oracle迁移的数据库与初始化数据库比较**
15. 表
16. 索引
17. 约束
18. 序列
19. 视图
20. 存储过程、函数
21. 触发器
22. **验证数据**
23. 表结构的一致性全量验证
24. 数据量的一致性-全量验证
25. 数据本身一致性-抽查30%验证
26. 数据编码的一致性-抽查30%验证