## 使用JDBC连接数据库

### 准备连接环境

修改数据库的pg\_hba.conf文件。

在GS\_HOME中查找pg\_hba.conf文件，本实验中数据库GS\_HOME设置的为/gaussdb/data/db1，实际操作中GS\_HOME地址可以查看安装时的配置文件：<PARAM name="dataNode1" value="**/gaussdb/data/db1**"/>。

**cd /gaussdb/data/db1**

**vi pg\_hba.conf**

输入“:90”找到对应位置，然后输入“i”切换到INSERT模式，将以下内容添加进pg\_hba.conf文件，添加后按下“ECS”键，退出INSERT模式，输入“:wq”后回车保存。

# IPv4 local connections:

host all all 127.0.0.1/32 trust

host all all 192.168.0.19/32 trust

**host all all 0.0.0.0/0 sha256**

# IPv6 local connections:

host all all ::1/128 trust

使用omm用户登陆，使用gs\_ctl将策略生效。

**su - omm**

**gs\_ctl reload -D /gaussdb/data/db1/**

返回结果为：

[2020-07-23 15:39:55.398][71828][][gs\_ctl]: gs\_ctl reload ,datadir is -D "/gaussdb/data/db1"

server signaled

注：如果之前没有启动过数据库，返回结果如下，继续操作即可：

[2021-03-18 15:37:57.305][3093][][gs\_ctl]: gs\_ctl reload ,datadir is /gaussdb/data/db1

[2021-03-18 15:37:57.306][3093][][gs\_ctl]: could not send reload signal (PID: 3946): No such process

使用omm用户登陆数据库创建dbuser用户并授权，最后退出数据库。

**gsql -d postgres -p 26000 -r**

连接数据库后，进入SQL命令界面。创建用户dbuser，密码为Gauss#3demo，并进行授权。

**CREATE USER dbuser IDENTIFIED BY 'Gauss#3demo';**

postgres=# **alter role dbuser createrole createdb;**

ALTER ROLE

postgres=# **\q**

修改数据库监听地址。

在GS\_HOME中，本实验中数据库GS\_HOME设置的为/gaussdb/data/db1。

**cd /gaussdb/data/db1**

**vi postgresql.conf**

输入“:60”找到对应位置，然后输入“i”切换到INSERT模式，将listen\_addresses的值修改成为\*，修改后按下“ECS”键，退出INSERT模式，输入“:wq”后回车保存。。

**#**listen\_addresses = '192.168.0.19' # what IP address(es) to listen on;

**listen\_addresses = '\*'**

修改完成后重启数据库生效（-D后面的数据库默认路径，需要根据实际情况进行修改）。

**gs\_ctl restart -D /gaussdb/data/db1/**

下载Java连接openGauss的驱动包，并将其导入对应的使用工具。

通过以下链接，下载驱动包

https://opengauss.obs.cn-south-1.myhuaweicloud.com/1.1.0/arm/openGauss-1.1.0-JDBC.tar.gz

假设文件存放在d:\Download目录下，并进行解压，解压后文件为“**postgresql.jar**”。

创建测试数据库demo。

使用gsql工具登陆数据库，并输入dbuser密码(如：**Gauss#3demo**)。

**gsql -d postgres -p 26000 -U dbuser -r**

创建数据库demo。

**create database demo ENCODING 'UTF8' template = template0;**

切换到demo数据库，并输入dbuser密码(如：**Gauss#3demo**)。

**\connect demo;**

创建名为demo的schema，并设置demo为当前的schema。

**CREATE SCHEMA demo;**

将默认搜索路径设为demo。

**SET search\_path TO demo;**

创建测试表websites。

**CREATE TABLE websites (**

**id int NOT NULL,**

**name char(20) NOT NULL DEFAULT '',**

**url varchar(255) NOT NULL DEFAULT '',**

**PRIMARY KEY (id)**

**);**

**COMMENT ON COLUMN websites.name IS '站点名称';**

插入数据。

**INSERT INTO websites VALUES**

**('1', 'openGauss', 'https://opengauss.org/zh/'),**

**('2', '华为云', 'https://www.huaweicloud.com/'),**

**('3', 'openEuler', 'https://openeuler.org/zh/'),**

**('4', '华为support中心', 'https://support.huaweicloud.com/');**

退出数据库

**\q**

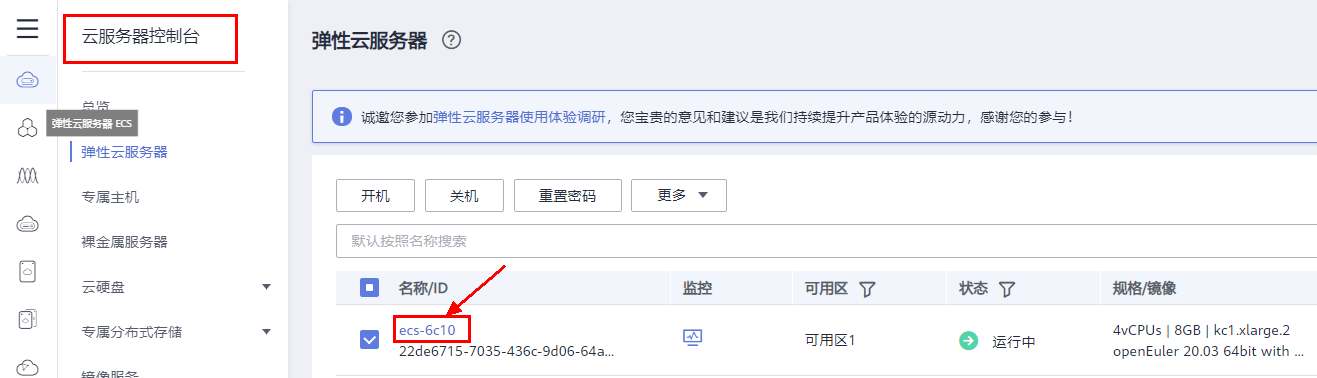
### 确定26000端口是否放开（虚拟机安装可以跳过这部）

打开华为云首页，登录后进入“控制台”，点击“弹性云服务器ECS”进入ECS列表。





在云服务器控制台找到安装数据库主机的ECS，点击查看基本信息，找到安全组。





点击进入安全组，选择“入方向规则”并“添加规则”，进行26000端口设置。



2.

1.



2.

1.

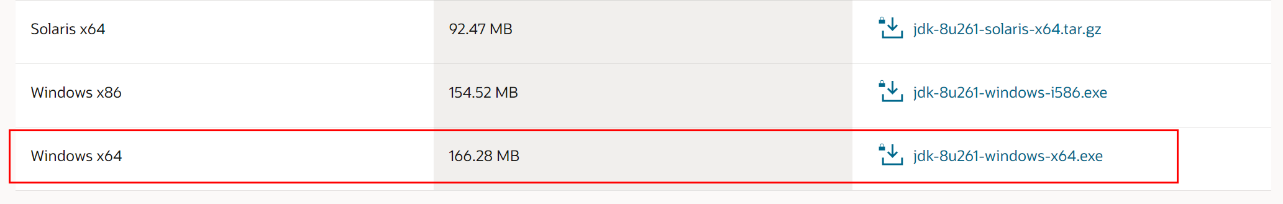
确定后，可以看到入网规则多了“TCP:26000”，如下图：



### 下载并安装JDK

下载JDK。

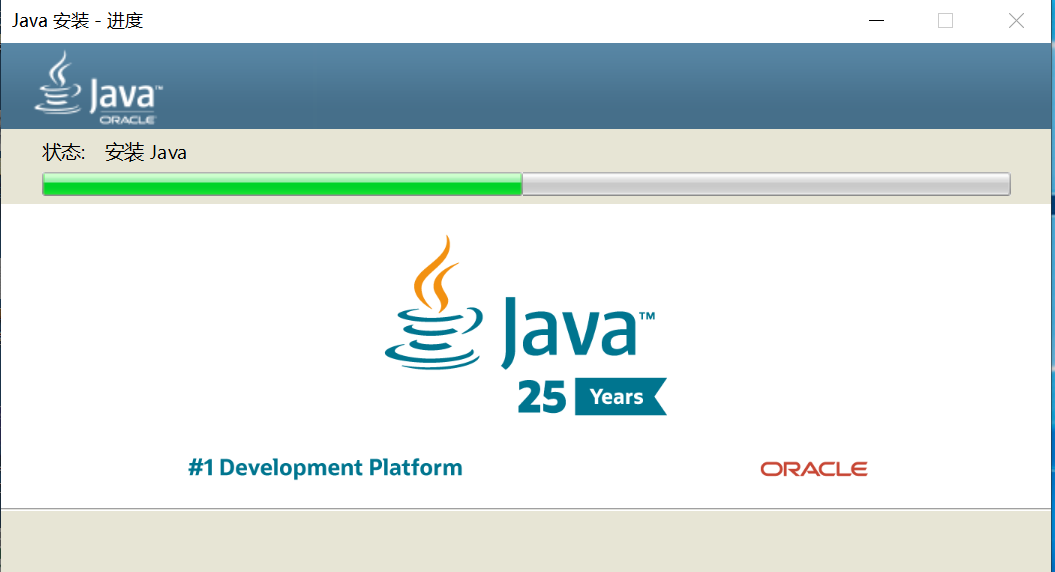
<https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>



C:\Users\swx941157\AppData\Roaming\eSpace_Desktop\UserData\swx941157\imagefiles\956D0976-2FE5-4327-BE30-8724B20B7D78.png

点击jdk-8u261-windows-x64.exe进行安装。

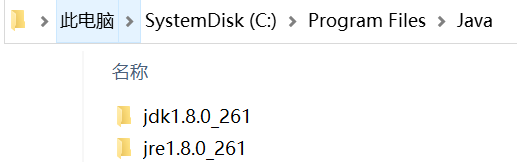
默认设置即可，出现安装进度。



如下显示表示安装成功：

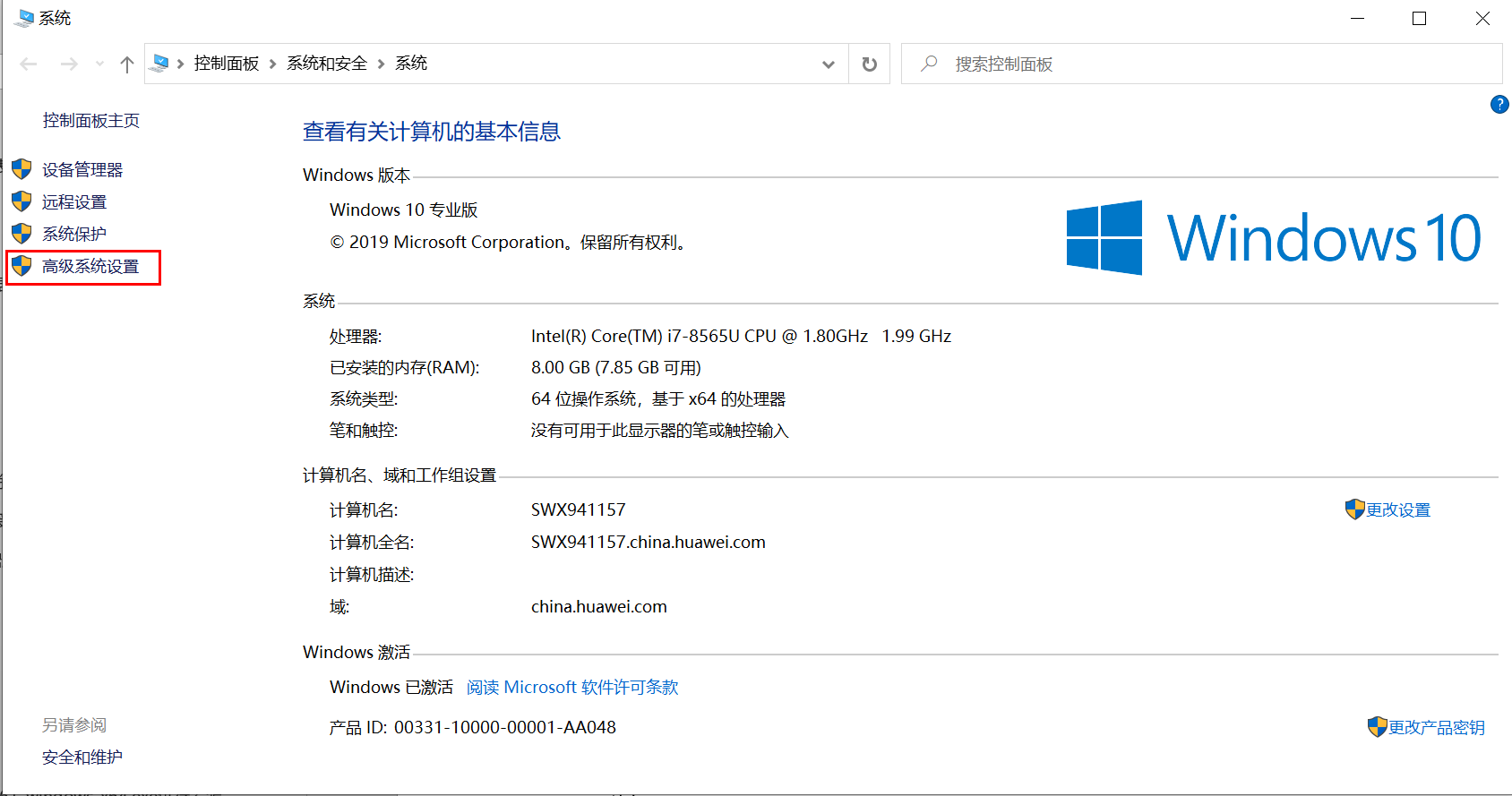


查看安装目录。



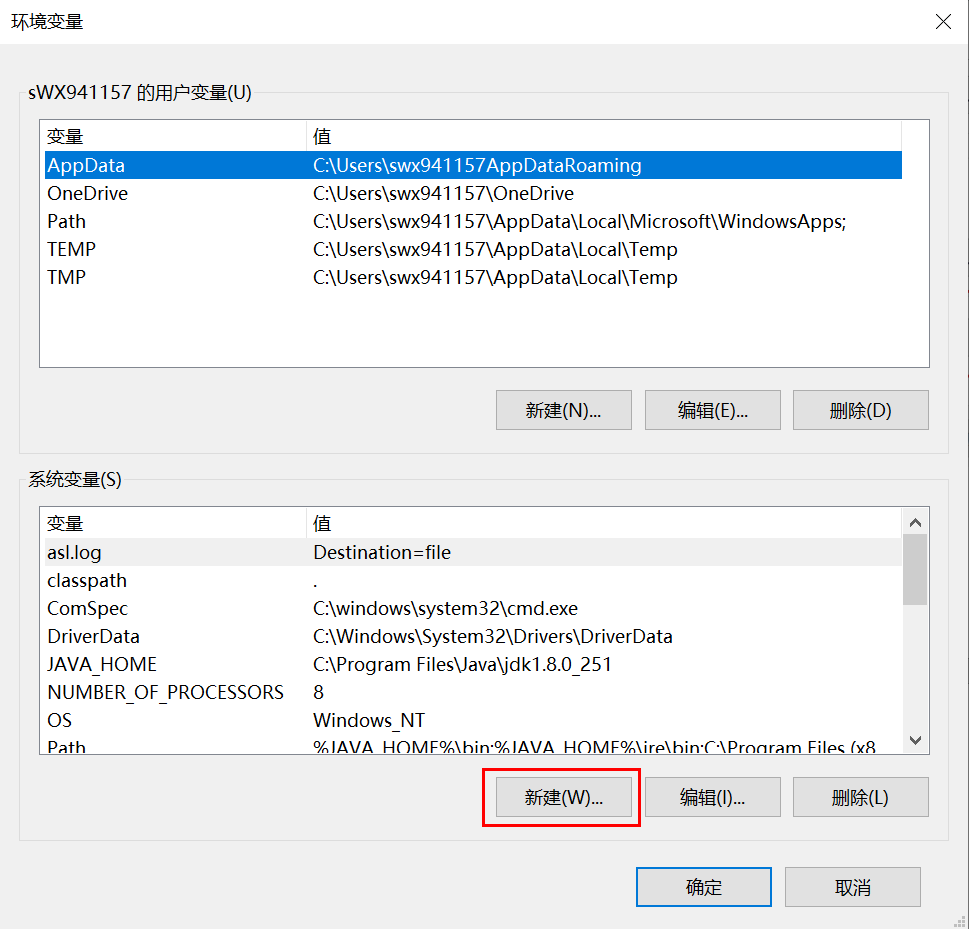
### 配置JDK环境变量

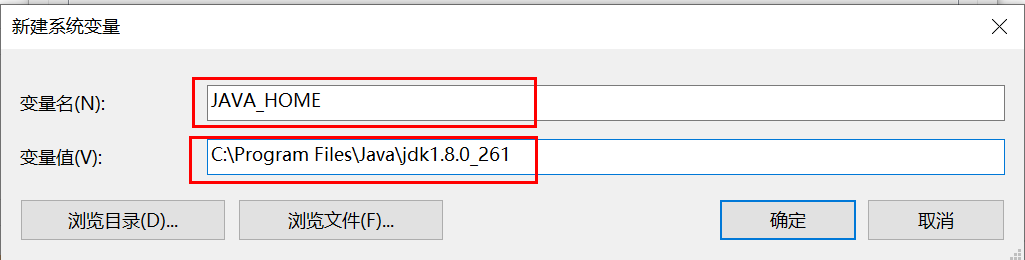
右击“此电脑”选择“属性”，点击“高级系统设置”。





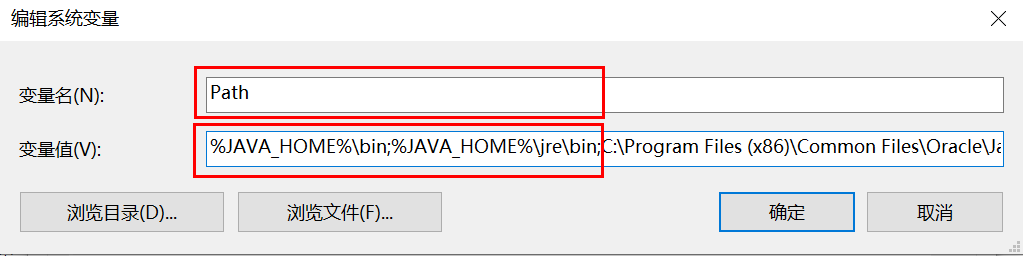
点击“环境变量”，新建系统变量“JAVA\_HOME”，输入JDK安装目录。





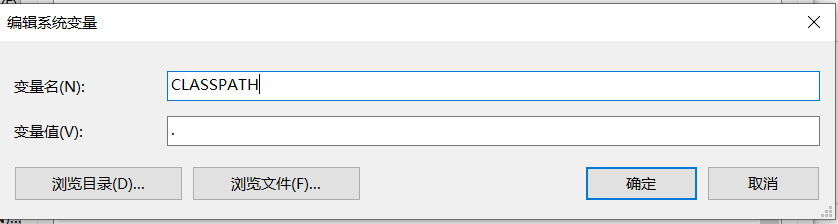
“C:\Program Files\Java\jdk1.8.0\_261”为JDK安装目录。

编辑系统变量“path”。



在变量值最后输入 %JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin;**（注意原来Path的变量值末尾有没有;号，如果没有，先输入；号再输入上面的代码）。**

新建系统变量“CLASSPATH”变量，输入“.” 即可。



系统变量配置完毕，查询检验是否配置成功，运行cmd 输入java -version（java和 -version之间有空格）。

C:\Users\xxxxx>**java -version**

如下所示，显示版本信息，则说明安装和配置成功。

java version "1.8.0\_261"

Java(TM) SE Runtime Environment (build 1.8.0\_261-b12)

Java HotSpot(TM) 64-Bit Server VM (build 25.261-b12, mixed mode)

### 连接openGauss并执行java代码

使用Java程序连接数据库并进行查询（注：请用户根据实际情况替换红字内容，修改jdbc:postgresql://**弹性公网IP**:26000/demo中弹性公网IP信息，USER = "**dbuser**"连接数据库的用户及密码PASS = "**Gauss#3demo**"）。

在d:\Download\中创建openGaussDemo.java文件，文件内容如下，注意红字部分要按照实际情况进行替换：

import java.sql.\*;

public class openGaussDemo {

static final String JDBC\_DRIVER = "org.postgresql.Driver";

static final String DB\_URL = "jdbc:postgresql://**弹性公网IP:**26000/demo?ApplicationName=app1";

// 数据库的用户名与密码，需要根据自己的设置

static final String USER = "**dbuser**";

static final String PASS = "**Gauss#3demo**";

public static void main(String[] args) {

Connection conn = null;

Statement stmt = null;

try{

// 注册 JDBC 驱动

Class.forName(JDBC\_DRIVER);

// 打开链接

System.out.println("连接数据库...");

conn = DriverManager.getConnection(DB\_URL,USER,PASS);

// 执行查询

System.out.println(" 实例化Statement对象...");

stmt = conn.createStatement();

String sql;

sql = "SELECT id, name, url FROM demo.websites";

ResultSet rs = stmt.executeQuery(sql);

// 展开结果集数据库

while(rs.next()){

// 通过字段检索

int id = rs.getInt("id");

String name = rs.getString("name");

String url = rs.getString("url");

// 输出数据

System.out.print("ID: " + id);

System.out.print(", 站点名称: " + name);

System.out.print(", 站点 URL: " + url);

System.out.print("\n");

}

// 完成后关闭

rs.close();

stmt.close();

conn.close();

}catch(SQLException se){

// 处理 JDBC 错误

se.printStackTrace();

}catch(Exception e){

// 处理 Class.forName 错误

e.printStackTrace();

}finally{

// 关闭资源

try{

if(stmt!=null) stmt.close();

}catch(SQLException se2){

}// 什么都不做

try{

if(conn!=null) conn.close();

}catch(SQLException se){

se.printStackTrace();

}

}

System.out.println("Goodbye!");

}

}

在安装Java的本机，打开cmd对Java程序编译后执行。

在cmd中，进入d:\Download\目录，先对Java程序进行编译（进入Java程序的目录）。

D:\Download> **javac -encoding utf-8 -cp d:\Download\postgresql.jar openGaussDemo.java**

再执行以下命令。

D:\Download> **java -cp .;D:/Download/postgresql.jar openGaussDemo**

执行结果。

执行结果如下：

连接数据库...

实例化Statement对象...

ID: 1, 站点名称: openGauss, 站点 URL: https://opengauss.org/zh/

ID: 2, 站点名称: 华为云, 站点 URL: https://www.huaweicloud.com/

ID: 3, 站点名称: openEuler, 站点 URL: https://openeuler.org/zh/

ID: 4, 站点名称: 华为support中心, 站点 URL: https://support.huaweicloud.com/

Goodbye!

## 使用ODBC连接数据库

### 准备连接环境

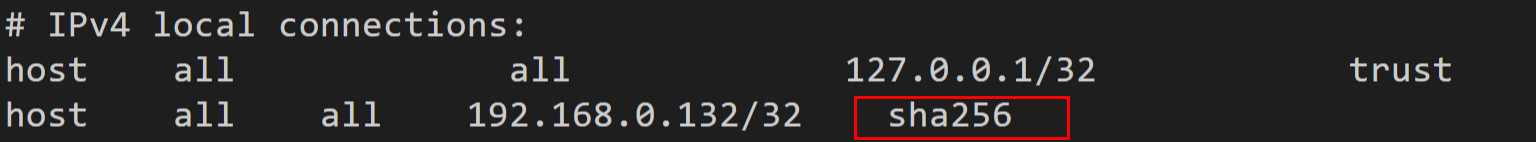
修改数据库的pg\_hba.conf文件。

在GS\_HOME中查找pg\_hba.conf文件，本实验中数据库GS\_HOME设置的为/gaussdb/data/db1，实际操作中GS\_HOME地址可以查看安装时的配置文件：<PARAM name="dataNode1" value="**/gaussdb/data/db1**"/>。

**cd /gaussdb/data/db1**

**vi pg\_hba.conf**

输入“:90”找到对应位置，然后输入“i”切换到INSERT模式，将“host all all 192.168.0.132/32 trust”修改为“host all all 192.168.0.132/32 **sha256**”。



将以下内容添加进pg\_hba.conf文件，完成后按下“ECS”键，退出INSERT模式，输入“:wq”后回车保存。

# IPv4 local connections:

host all all 127.0.0.1/32 trust

host all all 192.168.0.19/32 **sha256**

**host all all 0.0.0.0/0 sha256**

# IPv6 local connections:

host all all ::1/128 trust

使用omm用户登陆，使用gs\_ctl将策略生效。

**su - omm**

**gs\_ctl reload -D /gaussdb/data/db1/**

返回结果为：

[2020-07-23 15:39:55.398][71828][][gs\_ctl]: gs\_ctl reload ,datadir is -D "/gaussdb/data/db1"

server signaled

注：如果之前没有启动过数据库，返回结果如下，继续操作即可：

[2021-03-18 15:37:57.305][3093][][gs\_ctl]: gs\_ctl reload ,datadir is /gaussdb/data/db1

[2021-03-18 15:37:57.306][3093][][gs\_ctl]: could not send reload signal (PID: 3946): No such process

使用omm用户登陆数据库，给dbuser用户授权，并退出数据库。

**gsql -d postgres -p 26000 -r**

postgres=# **alter role dbuser sysadmin;**

ALTER ROLE

postgres=# **\q**

修改数据库监听地址。

在GS\_HOME中，本实验中数据库GS\_HOME设置的为/gaussdb/data/db1。

**cd /gaussdb/data/db1**

**vi postgresql.conf**

输入“:60”找到对应位置，然后输入“i”切换到INSERT模式，将listen\_addresses的值修改成为\*，修改后按下“ECS”键，退出INSERT模式，输入“:wq”后回车保存。。

**#**listen\_addresses = '192.168.0.19' # what IP address(es) to listen on;

**listen\_addresses = '\*'**

修改完成后重启数据库生效（-D后面的数据库默认路径，需要根据实际情况进行修改）。

**gs\_ctl restart -D /gaussdb/data/db1/**

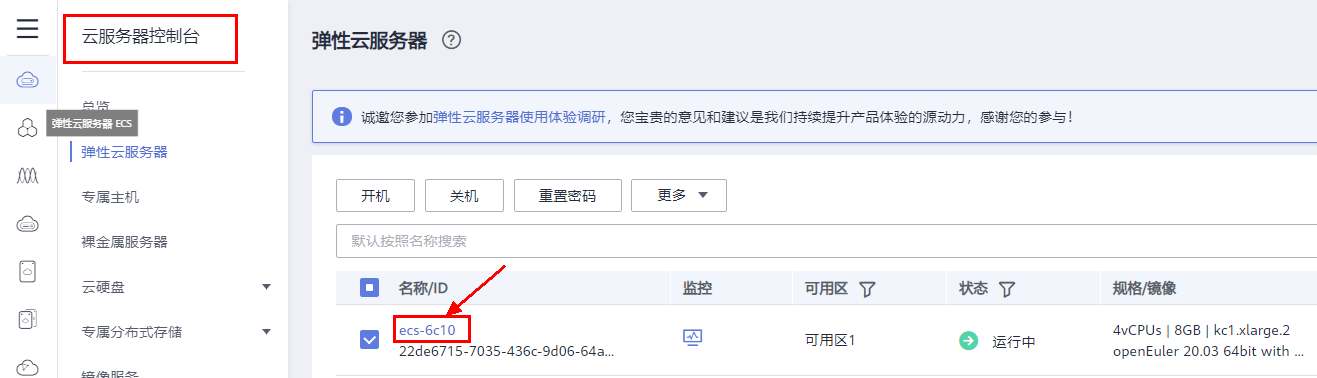
### 确定26000端口是否放开

打开华为云首页，登录后进入“控制台”，点击“弹性云服务器ECS”进入ECS列表。





在云服务器控制台找到安装数据库主机的ECS，点击查看基本信息，找到安全组。





点击进入安全组，选择“入方向规则”并“添加规则”，进行26000端口设置。



2.

1.



2.

1.

确定后，可以看到入网规则多了“TCP:26000”，如下图：



### 数据源安装配置

下载unixODBC源码包。

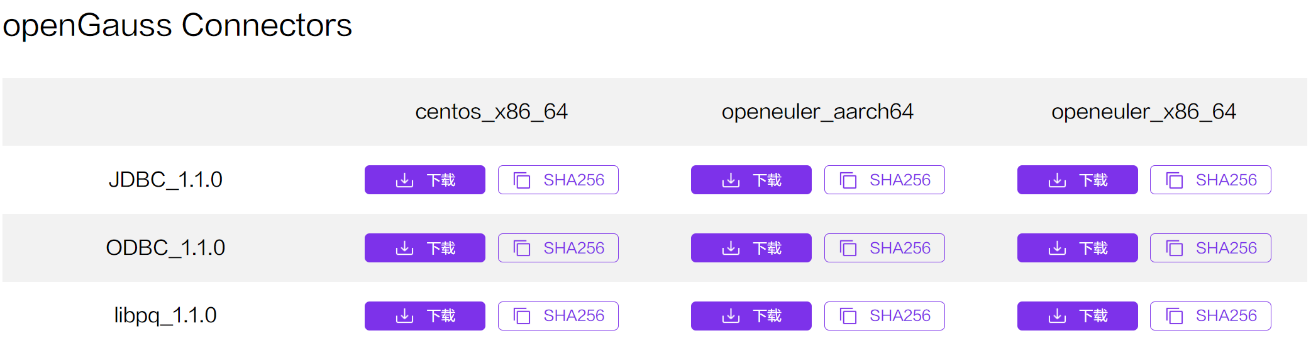
获取参考地址：<http://sourceforge.net/projects/unixodbc/files/unixODBC/2.3.0/unixODBC-2.3.0.tar.gz/download>



下载后的文件名为：unixODBC-2.3.0.tar.gz

下载客户端openGauss ODBC驱动包。

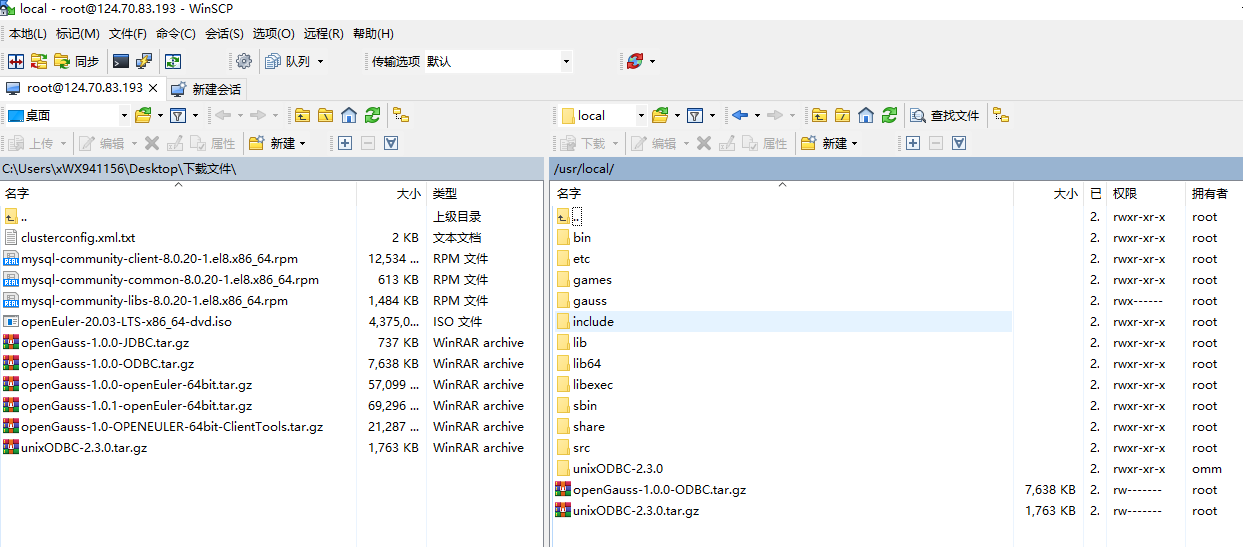
获取参考地址：https://opengauss.obs.cn-south-1.myhuaweicloud.com/1.1.0/arm/openGauss-1.1.0-ODBC.tar.gz



下载后的文件名为：openGauss-1.1.0-ODBC.tar.gz

上传unixODBC源码包、ODBC驱动包。

通过文件传输工具（如：WinSCP、XFTP等），将unixODBC源码包、ODBC驱动包二个包上传至数据库服务器/usr/local目录下。



安装unixODBC。

为了操作方便，可以通过SSH工具（如：putty、Xshell等）连接到数据库服务器，进行安装，具体操作如下：

先解压安装包。

[root@ecs-4e5b local]# **cd /usr/local**

[root@ecs-4e5b local]# **tar -zxvf unixODBC-2.3.0.tar.gz**

unixODBC-2.3.0/

unixODBC-2.3.0/ini/

unixODBC-2.3.0/ini/iniPropertyUpdate.c

unixODBC-2.3.0/ini/iniProperty.c

unixODBC-2.3.0/ini/iniOpen.c

……………………….

进行配置。

[root@ecs-4e5b local]# **cd unixODBC-2.3.0**

[root@db1 unixODBC-2.3.0]# **./configure --enable-gui=no --build=aarch64-unknown-linux-gnu**

checking for a BSD-compatible install... /usr/bin/install -c

checking whether build environment is sane... yes

checking for a thread-safe mkdir -p... /usr/bin/mkdir -p

checking for gawk... gawk

checking whether make sets $(MAKE)... yes

……………………….

进行make编译。

[root@db1 unixODBC-2.3.0]# **make**

make all-recursive

make[1]: Entering directory '/usr/local/unixODBC-2.3.0'

Making all in extras

make[2]: Entering directory '/usr/local/unixODBC-2.3.0/extras'

make[2]: Nothing to be done for 'all'.

make[2]: Leaving directory '/usr/local/unixODBC-2.3.0/extras'

……………………….

进行make install编译安装。

[root@db1 unixODBC-2.3.0]# **make install**

Making install in extras

make[1]: Entering directory '/usr/local/unixODBC-2.3.0/extras'

make[2]: Entering directory '/usr/local/unixODBC-2.3.0/extras'

make[2]: Nothing to be done for 'install-exec-am'.

make[2]: Nothing to be done for 'install-data-am'.

make[2]: Leaving directory '/usr/local/unixODBC-2.3.0/extras'

make[1]: Leaving directory '/usr/local/unixODBC-2.3.0/extras'

……………………….

将openGauss-1.0.0-ODBC.tar.gz解压到“/usr/local/lib”目录下。

[root@db1 local]# **cd /usr/local**

[root@db1 local]# **tar -zxvf /usr/local/openGauss-1.1.0-ODBC.tar.gz -C /usr/local/lib**

./lib/

./lib/libodbcinst.so.2

./lib/libpq.so.5

./lib/libpgport\_tool.so

./lib/libcom\_err\_gauss.so

./lib/libodbcinst.so

./lib/libcom\_err\_gauss.so.3.0

./lib/libkrb5\_gauss.so

……………………

将openGauss-1.0.0-ODBC.tar.gz解压后lib目录中的库拷贝到“/usr/local/lib”目录下。

[root@db1 lib]# **cd /usr/local/lib/odbc/lib**

[root@db1 lib]# **ll**

total 1.6M

-rwxr-xr-x 1 root root 1.3K Oct 12 02:12 psqlodbca.la

-rwxr-xr-x 1 root root 766K Oct 12 02:12 psqlodbca.so

-rwxr-xr-x 1 root root 1.3K Oct 12 02:12 psqlodbcw.la

-rwxr-xr-x 1 root root 772K Oct 12 02:12 psqlodbcw.so

[root@db1 lib]# **cp \*.\* /usr/local/lib**

[root@db1 lib]#

配置ODBC驱动文件。

在“/usr/local/etc/odbcinst.ini”文件中追加以下内容。

[GaussMPP]

Driver64=/usr/local/lib/psqlodbcw.so

setup=/usr/local/lib/psqlodbcw.so

具体如下：

[root@db1 etc]# **vi /usr/local/etc/odbcinst.ini**

[GaussMPP]

Driver64=/usr/local/lib/psqlodbcw.so

setup=/usr/local/lib/psqlodbcw.so

配置数据源文件。

在“/usr/local/etc/odbc.ini ”文件中追加以下内容。

[MPPODBC] (数据源名称)

Driver=GaussMPP (驱动名称)

Servername=10.10.0.13（数据库Server IP）---本实验用的是ECS私有IP

Database=postgres （数据库名）

Username=omm （数据库用户名）

Password= （数据库用户密码）

Port=8000 （数据库监听端口）

Sslmode=allow

具体如下：

[root@db1 etc]# **vi /usr/local/etc/odbc.ini**

[MPPODBC]

Driver=GaussMPP

Servername=**192.168.0.208**

Database=postgres

Username=dbuser

Password=Gauss#3demo

Port=26000

Sslmode=allow

在客户端配置环境变量。

本实验的客户端与数据库在同一台服务器。

在root的.bashrc文件中追加以下内容。

export LD\_LIBRARY\_PATH=/usr/local/lib/:/usr/local/lib/lib:$LD\_LIBRARY\_PATH

export ODBCSYSINI=/usr/local/etc

export ODBCINI=/usr/local/etc/odbc.ini

具体如下：

[root@db1 etc]# **cd ~**

[root@db1 ~]# **vim ~/.bashrc**

# .bashrc

# User specific aliases and functions

alias rm='rm -i'

alias cp='cp -i'

alias mv='mv -i'

# Source global definitions

if [ -f /etc/bashrc ]; then

. /etc/bashrc

fi

export LD\_LIBRARY\_PATH=/usr/local/lib/:/usr/local/lib/lib:$LD\_LIBRARY\_PATH

export ODBCSYSINI=/usr/local/etc

export ODBCINI=/usr/local/etc/odbc.ini

使用环境变量生效。

[root@db1 ~]# **source ~/.bashrc**

测试数据源配置

执行isql -v MPPODBC (数据源名称)命令。

如果显示如下信息，表明配置正确，连接成功。

[root@db1 ~]# **isql -v MPPODBC**

+---------------------------------------+

| Connected! |

| |

| sql-statement |

| help [tablename] |

| quit |

| |

+---------------------------------------+

SQL>

### 连接openGauss并执行C代码

使用C程序连接数据库并进行查询（注：请用户根据实际情况替换红字内容，修改这里的“**dbuser**”与“**Gauss#3demo**”、 **MPPODBC**分别表示连接数据库的用户名和用户密码、数据源名称，请根据实际情况修改）。

在**/usr/local/lib**目录下通过 vi DBtest.c 命令来创建一个C文件（DBtest.c），并将以下内容添加至DBtest.c文件中。

文件内容如如下，注意红字部分要按照实际情况进行替换：

// 此示例演示如何通过ODBC方式获取openGauss中的数据。

// DBtest.c (compile with: libodbc.so)

#include <stdlib.h>

#include <stdio.h>

#include <sqlext.h>

#ifdef WIN32

#include <windows.h>

#endif

SQLHENV V\_OD\_Env; // Handle ODBC environment

SQLHSTMT V\_OD\_hstmt; // Handle statement

SQLHDBC V\_OD\_hdbc; // Handle connection

char typename[100];

SQLINTEGER value = 600;

SQLINTEGER V\_OD\_erg,V\_OD\_buffer,V\_OD\_err,V\_OD\_id;

int main(int argc,char \*argv[])

{

// 1. 申请环境句柄

V\_OD\_erg = SQLAllocHandle(SQL\_HANDLE\_ENV,SQL\_NULL\_HANDLE,&V\_OD\_Env);

if ((V\_OD\_erg != SQL\_SUCCESS) && (V\_OD\_erg != SQL\_SUCCESS\_WITH\_INFO))

{

printf("Error AllocHandle\n");

exit(0);

}

// 2. 设置环境属性（版本信息）

SQLSetEnvAttr(V\_OD\_Env, SQL\_ATTR\_ODBC\_VERSION, (void\*)SQL\_OV\_ODBC3, 0);

// 3. 申请连接句柄

V\_OD\_erg = SQLAllocHandle(SQL\_HANDLE\_DBC, V\_OD\_Env, &V\_OD\_hdbc);

if ((V\_OD\_erg != SQL\_SUCCESS) && (V\_OD\_erg != SQL\_SUCCESS\_WITH\_INFO))

{

SQLFreeHandle(SQL\_HANDLE\_ENV, V\_OD\_Env);

exit(0);

}

// 4. 设置连接属性

SQLSetConnectAttr(V\_OD\_hdbc, SQL\_ATTR\_AUTOCOMMIT, SQL\_AUTOCOMMIT\_ON, 0);

// 5. 连接数据源，这里的“**dbuser**”与“**Gauss#3demo**”、 **MPPODBC**分别表示连接数据库的用户名和用户密码、数据源名称，请根据实际情况修改。

V\_OD\_erg = SQLConnect(V\_OD\_hdbc, (SQLCHAR\*) "**MPPODBC**", SQL\_NTS,

(SQLCHAR\*) "**dbuser**", SQL\_NTS, (SQLCHAR\*) "**Gauss#3demo**", SQL\_NTS);

if ((V\_OD\_erg != SQL\_SUCCESS) && (V\_OD\_erg != SQL\_SUCCESS\_WITH\_INFO))

{

printf("Error SQLConnect %d\n",V\_OD\_erg);

SQLFreeHandle(SQL\_HANDLE\_ENV, V\_OD\_Env);

exit(0);

}

printf("Connected !\n");

// 6. 设置语句属性

SQLSetStmtAttr(V\_OD\_hstmt,SQL\_ATTR\_QUERY\_TIMEOUT,(SQLPOINTER \*)3,0);

// 7. 申请语句句柄

SQLAllocHandle(SQL\_HANDLE\_STMT, V\_OD\_hdbc, &V\_OD\_hstmt);

// 8. 直接执行SQL语句。

SQLExecDirect(V\_OD\_hstmt,"drop table IF EXISTS customer\_t1",SQL\_NTS);

SQLExecDirect(V\_OD\_hstmt,"CREATE TABLE customer\_t1(c\_customer\_sk INTEGER, c\_customer\_name VARCHAR(32))",SQL\_NTS);

SQLExecDirect(V\_OD\_hstmt,"insert into customer\_t1 values(25,'li')",SQL\_NTS);

// 9. 准备执行

SQLPrepare(V\_OD\_hstmt,"insert into customer\_t1 values(?)",SQL\_NTS);

// 10. 绑定参数

SQLBindParameter(V\_OD\_hstmt,1,SQL\_PARAM\_INPUT,SQL\_C\_SLONG,SQL\_INTEGER,0,0,

&value,0,NULL);

// 11. 执行准备好的语句

SQLExecute(V\_OD\_hstmt);

SQLExecDirect(V\_OD\_hstmt,"select c\_customer\_sk from customer\_t1",SQL\_NTS);

// 12. 获取结果集某一列的属性

SQLColAttribute(V\_OD\_hstmt,1,SQL\_DESC\_TYPE,typename,200,NULL,NULL);

printf("SQLColAtrribute %s\n",typename);

// 13. 绑定结果集

SQLBindCol(V\_OD\_hstmt,1,SQL\_C\_SLONG, (SQLPOINTER)&V\_OD\_buffer,150,

(SQLLEN \*)&V\_OD\_err);

// 14. 通过SQLFetch取结果集中数据

V\_OD\_erg=SQLFetch(V\_OD\_hstmt);

// 15. 通过SQLGetData获取并返回数据。

while(V\_OD\_erg != SQL\_NO\_DATA)

{

SQLGetData(V\_OD\_hstmt,1,SQL\_C\_SLONG,(SQLPOINTER)&V\_OD\_id,0,NULL);

printf("SQLGetData ----ID = %d\n",V\_OD\_id);

V\_OD\_erg=SQLFetch(V\_OD\_hstmt);

};

printf("Done !\n");

// 16. 断开数据源连接并释放句柄资源

SQLFreeHandle(SQL\_HANDLE\_STMT,V\_OD\_hstmt);

SQLDisconnect(V\_OD\_hdbc);

SQLFreeHandle(SQL\_HANDLE\_DBC,V\_OD\_hdbc);

SQLFreeHandle(SQL\_HANDLE\_ENV, V\_OD\_Env);

return(0);

}

编译DBtest.c代码。

通过gcc -lodbc -o Mytest DBtest.c 命令来编译，编译后生成一个Mytest执行文件

[root@db1 lib]# **gcc -lodbc -o Mytest DBtest.c**

DBtest.c: In function ‘main’:

DBtest.c:34:57: warning: passing argument 3 of ‘SQLSetConnectAttr’ makes pointer from integer without a cast [-Wint-conversion]

SQLSetConnectAttr(V\_OD\_hdbc, SQL\_ATTR\_AUTOCOMMIT, SQL\_AUTOCOMMIT\_ON, 0);

^~~~~~~~~~~~~~~~~

In file included from /usr/local/include/sqlext.h:43:0,

from DBtest.c:5:

/usr/local/include/sql.h:765:24: note: expected ‘SQLPOINTER {aka void \*}’ but argument is of type ‘long unsigned int’

SQLRETURN SQL\_API SQLSetConnectAttr(SQLHDBC ConnectionHandle,

^~~~~~~~~~~~~~~~~

[root@db1 lib]# **ll Mytest**

-rwx------ 1 root root 71K Nov 6 18:04 Mytest

执行连接并操作数据库。

执行结果如下：

[root@db1 lib]# **./Mytest**

Connected !

SQLColAtrribute

SQLGetData ----ID = 25

SQLGetData ----ID = 600

Done !

[root@db1 lib]#

执行成功后，会在dbuser用户下创建一张customer\_t1表，并在表中插入二条数据。

登录数据库进行验证。

先切换到OMM用户下。

[root@db1 lib]# **su - omm**

Last login: Fri Nov 6 16:20:14 CST 2020 on pts/0

Welcome to 4.19.90-2003.4.0.0036.oe1.aarch64

System information as of time: Fri Nov 6 18:10:39 CST 2020

System load: 0.00

Processes: 127

Memory used: 8.7%

Swap used: 0.0%

Usage On: 12%

IP address: 192.168.0.208

Users online: 2

然后登录数据库。

[omm@db1 ~]$ **gsql -d postgres -p 26000 -r**

gsql ((openGauss 1.0.1 build 13b34b53) compiled at 2020-10-12 02:01:33 commit 0 last mr )

Non-SSL connection (SSL connection is recommended when requiring high-security)

Type "help" for help.

进行查询验证。

postgres=# **select \* from dbuser.customer\_t1;**

c\_customer\_sk | c\_customer\_name

---------------+-----------------

25 | li

600 |

(2 rows)

postgres=#

## 实验小结

本实验通过ER模型加深对数据库的理解和数据库设计知识的掌握，通过SQL语句的练习，使学员熟练掌握SQL语法。

# 附录一：Windows下配置ODBC数据源

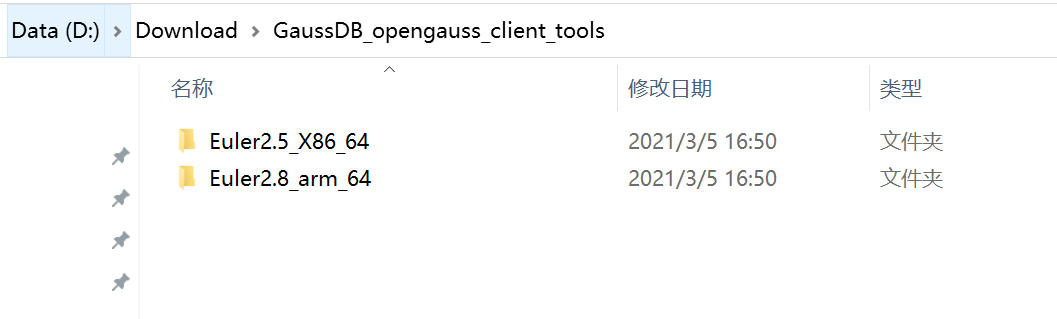
Windows操作系统自带ODBC数据源管理器，无需用户手动安装管理器便可直接进行配置。

## 操作步骤

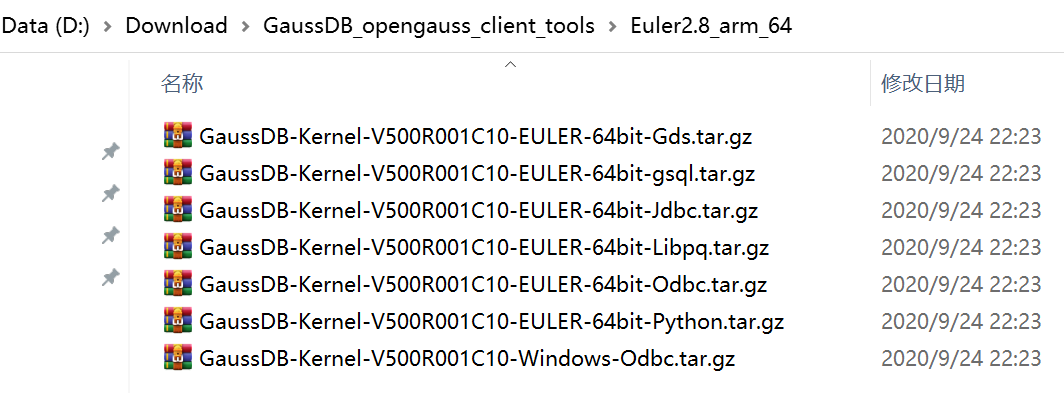
下载客户端GaussDB(for openGauss)驱动程序并进行安装：

下载地址：<https://dbs-download.obs.cn-north-1.myhuaweicloud.com/rds/GaussDB_opengauss_client_tools.zip>

在本地（例如D:/download）下载ZIP文件后进行解压缩，解压缩后文件如下。



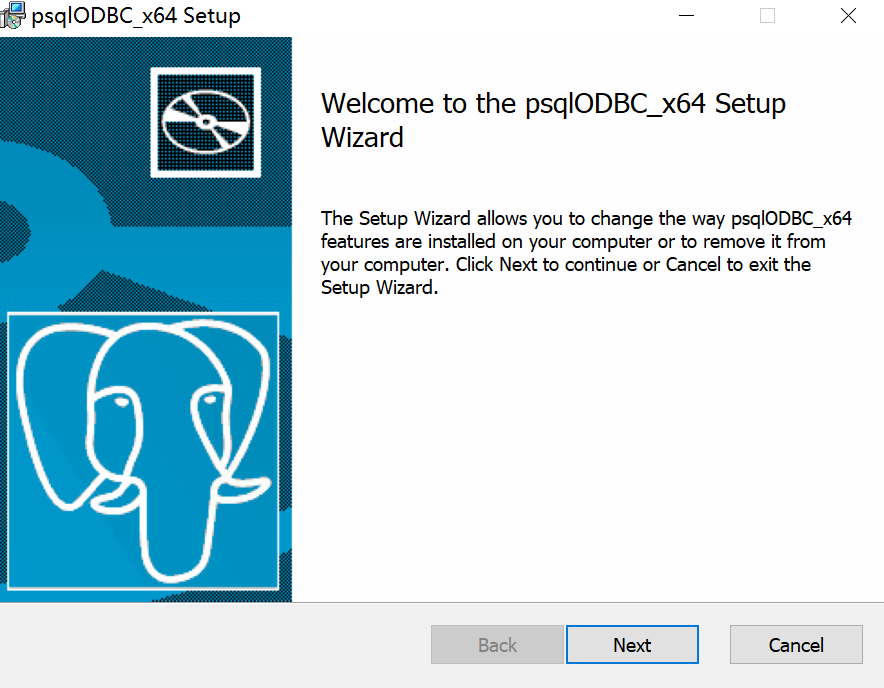
由于本实验openGauss安装在ECS（openEuler ARM）上，所以进入Euler2.8\_arm\_64文件夹，显示如下：



解压缩GaussDB-Kernel-V500R001C10-Windows-Odbc.tar.gz文件，显示如下：



点击psqlodbc\_x86.msi进行安装：



默认设置，直到安装完毕。

打开驱动管理器。

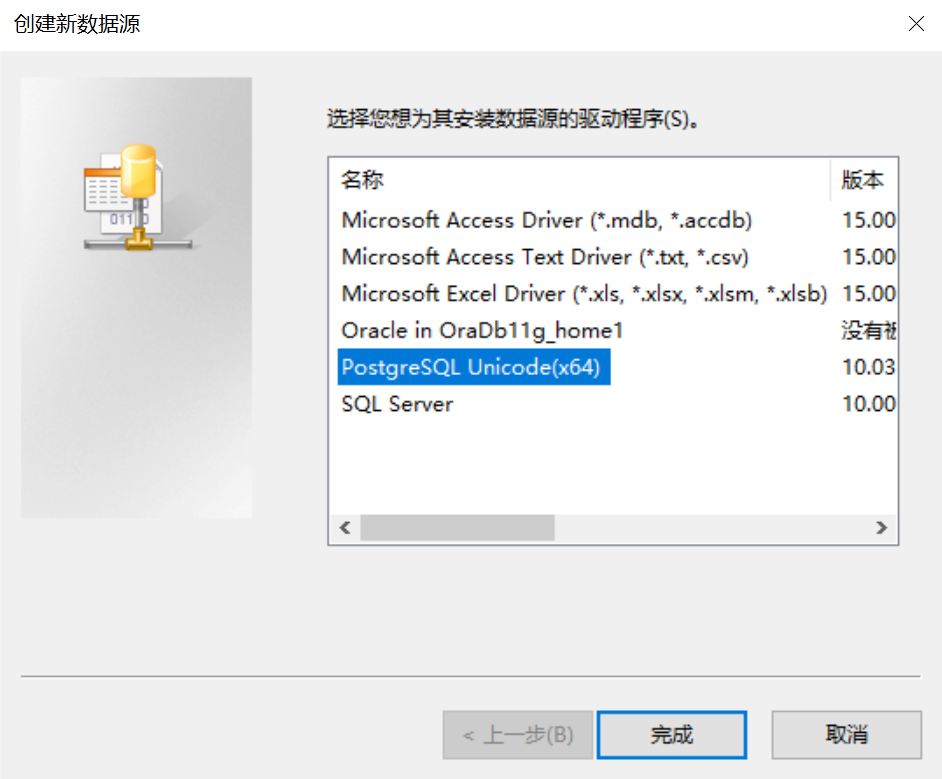
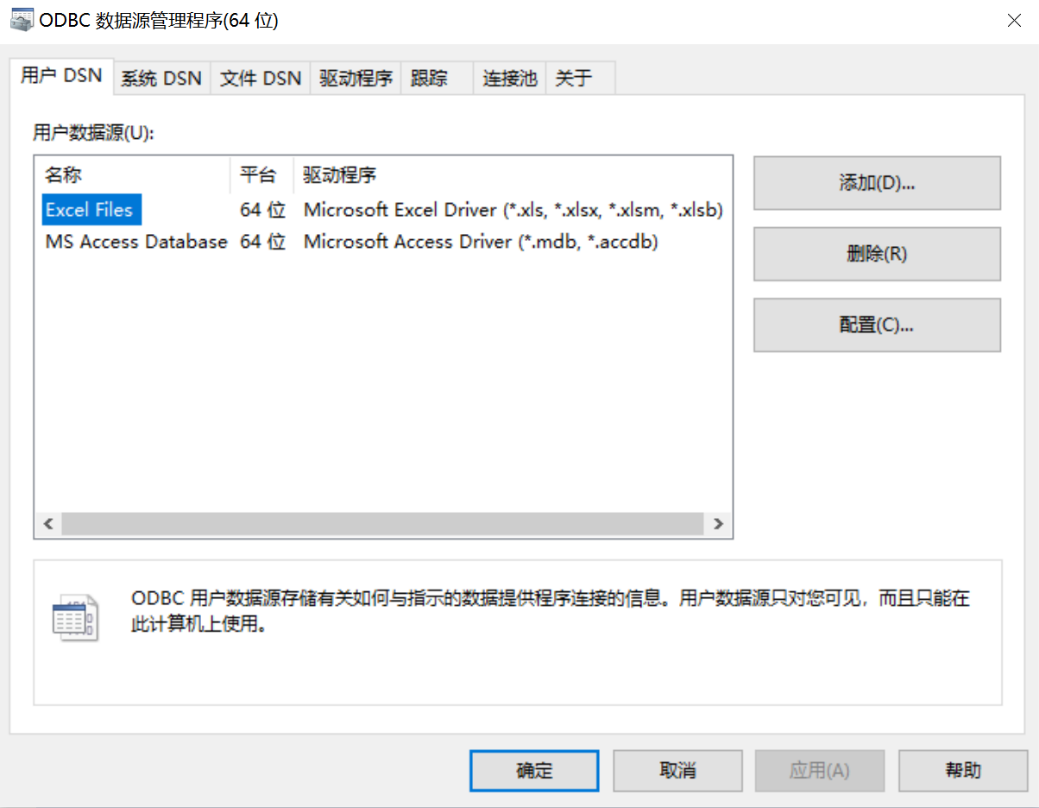
在配置数据源时，请使用对应的驱动管理器（假设操作系统安装盘符为C盘，如果是其他盘符，请对路径做相应修改）：

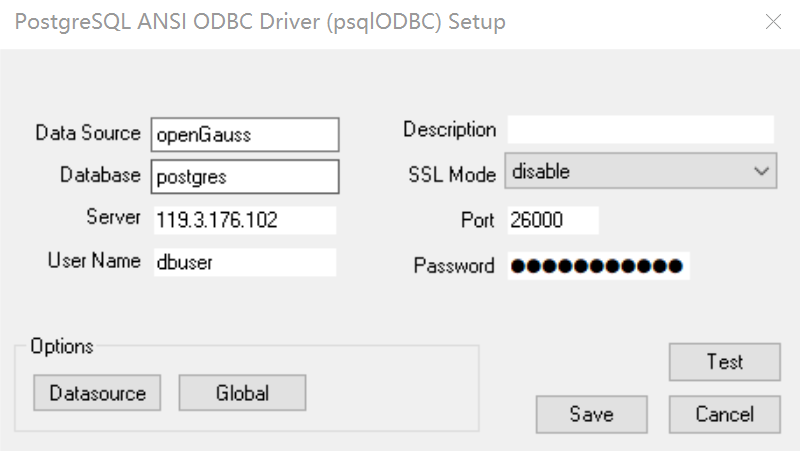
**64**位操作系统上进行**64**位程序开发，安装**64**位驱动程序后，使用**64**位的驱动管理器：**C:\Windows\System32\odbcad32.exe** 或者直接使用“控制面板 > 管理工具 > ODBC 数据源(64 位)。



配置数据源。

在打开的驱动管理器上，选择“用户DSN > 添加 > PostgreSQL Unicode(x64)”，然后进行配置：





Data Source:openGauss （数据源名称，可以自定义）。

Database:postgres (需要连接的Datebase名称)。

Server：**openGauss数据库服务器的公网IP，请根据实际情况填写。**

Pot:26000 (端口号)。

User Name:dbuser （连接数据库的用户名，不能使用omm用户，需要在数据库中创建）

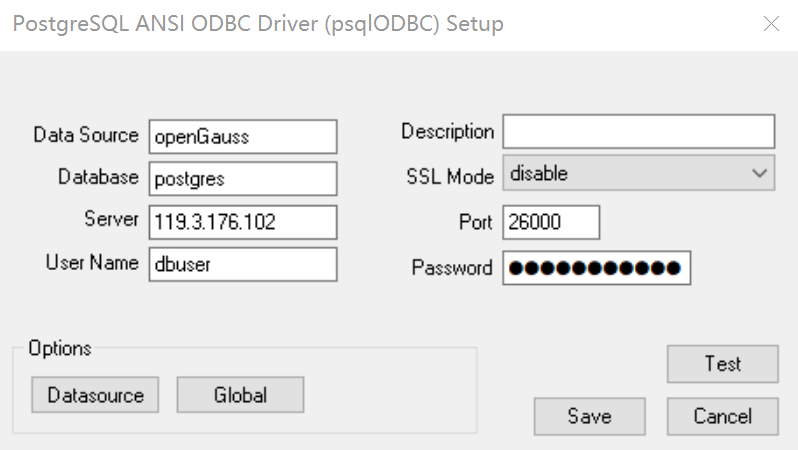
Password：dbuser用户的密码，请根据实际情况填写。

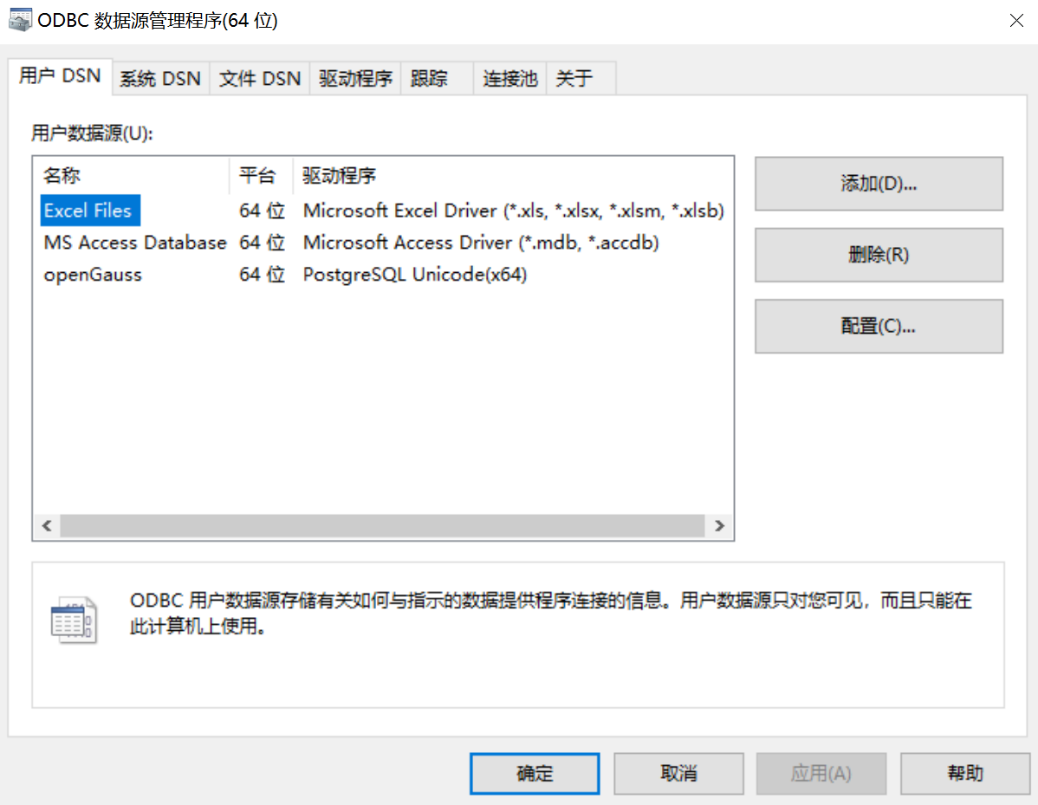
验证并保存设置。

点击Test，显示 Connection successful 表示设置成功：



点击保存：





配置成功！

# 附录二：openGauss数据库基本操作

## 查看数据库对象

* 查看帮助信息：

postgres=# \?

* 切换数据库：

*postgres*=# \c dbname

* 列举数据库：

使用\l元命令查看数据库系统的数据库列表。

postgres=# \l

使用如下命令通过系统表pg\_database查询数据库列表。

*postgres*=# SELECT datname FROM pg\_database;

* 列举表：

postgres=# \dt

* 列举所有表、视图和索引：

postgres=# \d+

* 使用gsql的\d+命令查询表的属性。

*postgres*=# \d+ tablename

* 查看表结构：

*postgres*=# \d tablename

* 列举schema：

postgres=# \dn

* 查看索引：

postgres=# \di

* 查询表空间：

使用gsql程序的元命令查询表空间。

postgres=# \db

检查pg\_tablespace系统表。如下命令可查到系统和用户定义的全部表空间。

*postgres*=# SELECT spcname FROM pg\_tablespace;

* 查看数据库用户列表：

*postgres*=# SELECT \* FROM pg\_user;

* 要查看用户属性：

*postgres*=# SELECT \* FROM pg\_authid;

* 查看所有角色：

*postgres*=# SELECT \* FROM PG\_ROLES;

## 其他操作

* 查看openGauss支持的所有SQL语句。

postgres=#\h

* 切换数据库：

*postgres*=# \c dbname

* 切换用户：

*postgres*=# \c – username

* 退出数据库：

postgres=# \q