《数据库》

Windows下应用程序连接GaussDB(for openGauss)

实验指导



华为技术有限公司

目录

[前 言 3](#_Toc99037298)

[简介 3](#_Toc99037299)

[实验环境说明 3](#_Toc99037300)

[1 应用程序连接前准备 4](#_Toc99037301)

[1.1 购买GaussDB(for openGauss)实例 4](#_Toc99037302)

[1.2 绑定弹性公网IP及开放8000端口 4](#_Toc99037303)

[2 使用python连接数据库 10](#_Toc99037304)

[2.1 下载并安装Python 10](#_Toc99037305)

[2.2 连接准备 12](#_Toc99037306)

[2.3 连接和使用数据库 13](#_Toc99037307)

[3 使用JDBC连接数据库 15](#_Toc99037308)

[3.1 下载并安装JDK 15](#_Toc99037309)

[3.2 配置JDK环境变量 16](#_Toc99037310)

[3.3 连接准备 20](#_Toc99037311)

[3.3.1 准备实验数据 20](#_Toc99037312)

[3.3.2 下载GaussDB(for openGauss)的Java驱动包 23](#_Toc99037313)

[3.4 连接GaussDB(for openGauss)并执行Java代码 24](#_Toc99037314)

[4 使用ODBC连接数据库 27](#_Toc99037315)

[4.1 加载GaussDB(for openGauss) ODBC驱动包 27](#_Toc99037316)

[4.2 配置数据源 28](#_Toc99037317)

[4.3 执行示例代码连接并操作数据库 30](#_Toc99037318)

前 言

简介

本实验指导书介绍使用驱动实现Python、Java、C/C++连接GaussDB(for openGauss)的方法，主要目的是为了让读者学习GaussDB(for openGauss)数据库的应用程序连接方法。

实验环境说明

* 组网说明

本实验环境为华为云GaussDB(for openGauss)数据库。

* 设备介绍

为了满足本实验需要，建议实验环境采用以下配置：

设备明细表

|  |  |  |
| --- | --- | --- |
| 设备名称 | 设备型号 | 软件版本 |
| 数据库 | GaussDB(for openGauss) 8 核 | 64 GB | GaussDB(for openGauss)服务 |

* 实验环境预置

购买GaussDB(for openGauss)数据库实例。

绑定弹性公网IP及开放8000端口。

# 应用程序连接前准备

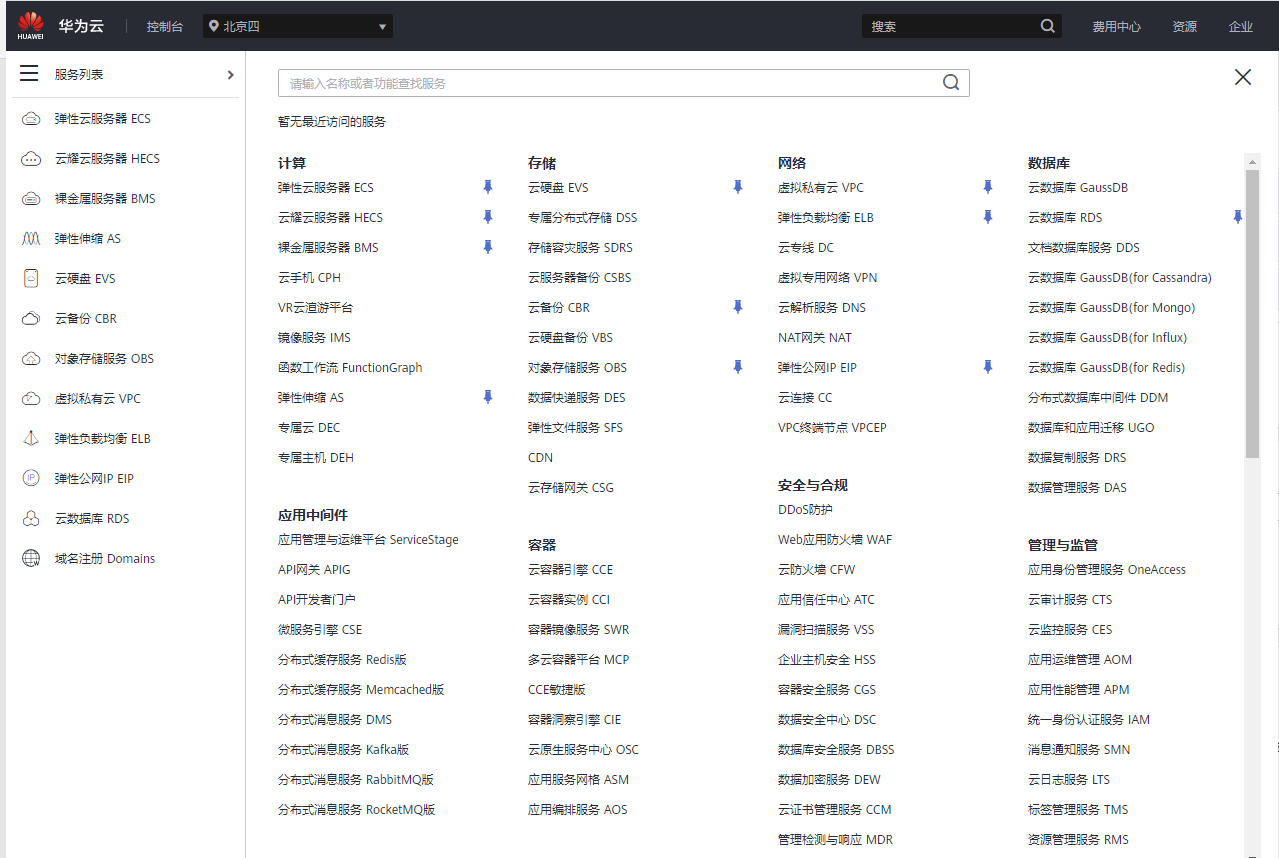
## 购买GaussDB(for openGauss)实例

如果已经购买了GaussDB(for openGauss)数据库可以跳过此节。如果还未购买，请参照《GaussDB(for openGauss)数据库实验环境搭建指导手册》进行购买。

## 绑定弹性公网IP及开放8000端口

如果GaussDB(for openGauss)数据库实例已经绑定弹性公网IP，且开放了8000端口，可以跳过此节。如果没有绑定，请参照如下步骤进行绑定。

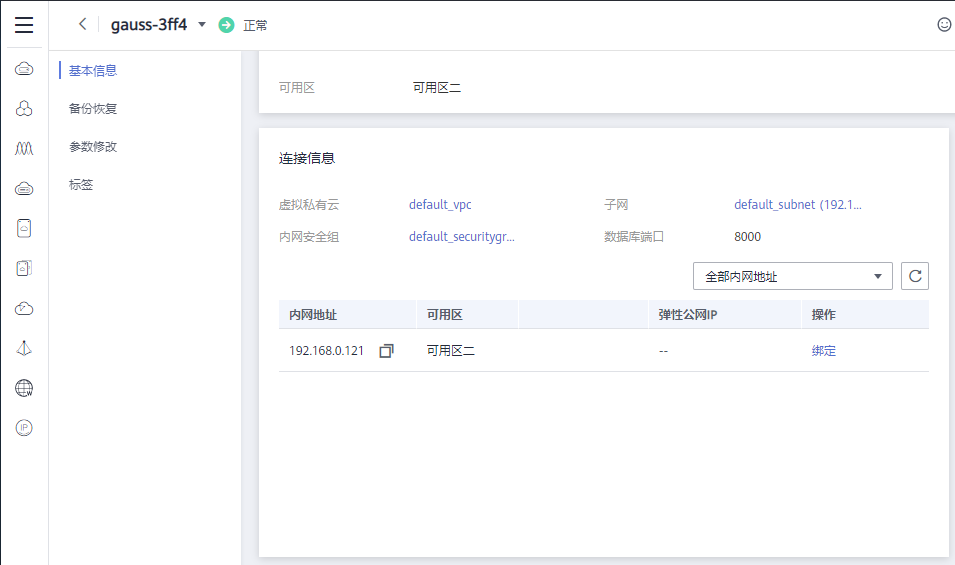
单击打开服务列表下，然后选择”数据库”下的云数据库GaussDB，具体如下：



进入数据库服务列表，选择GaussDB(for openGauss)。



单击实例名称，进入实例基本信息页面。



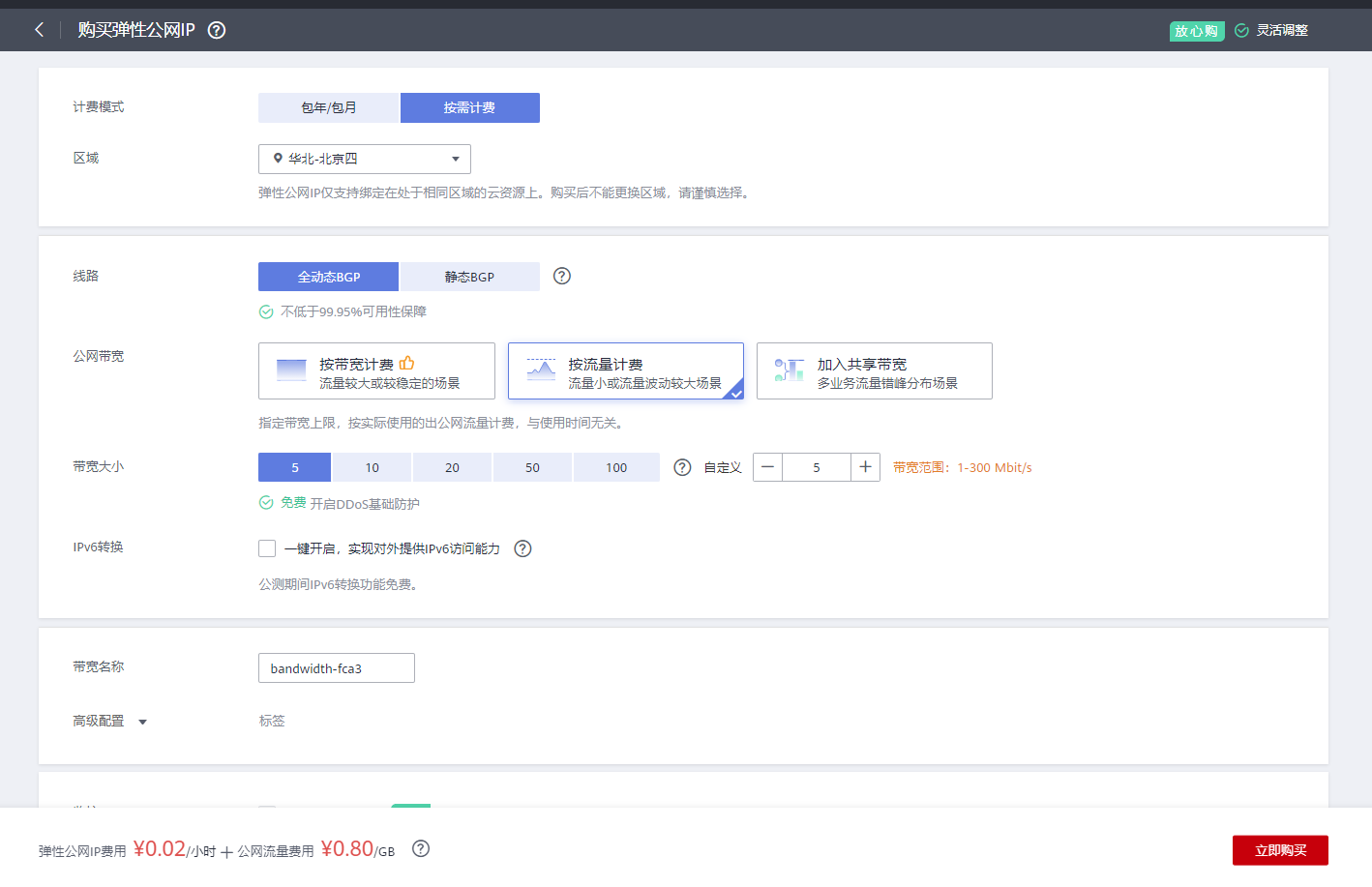
如果弹性公网IP没有绑定的话，请单击连接信息中的“绑定”按钮来进行绑定。



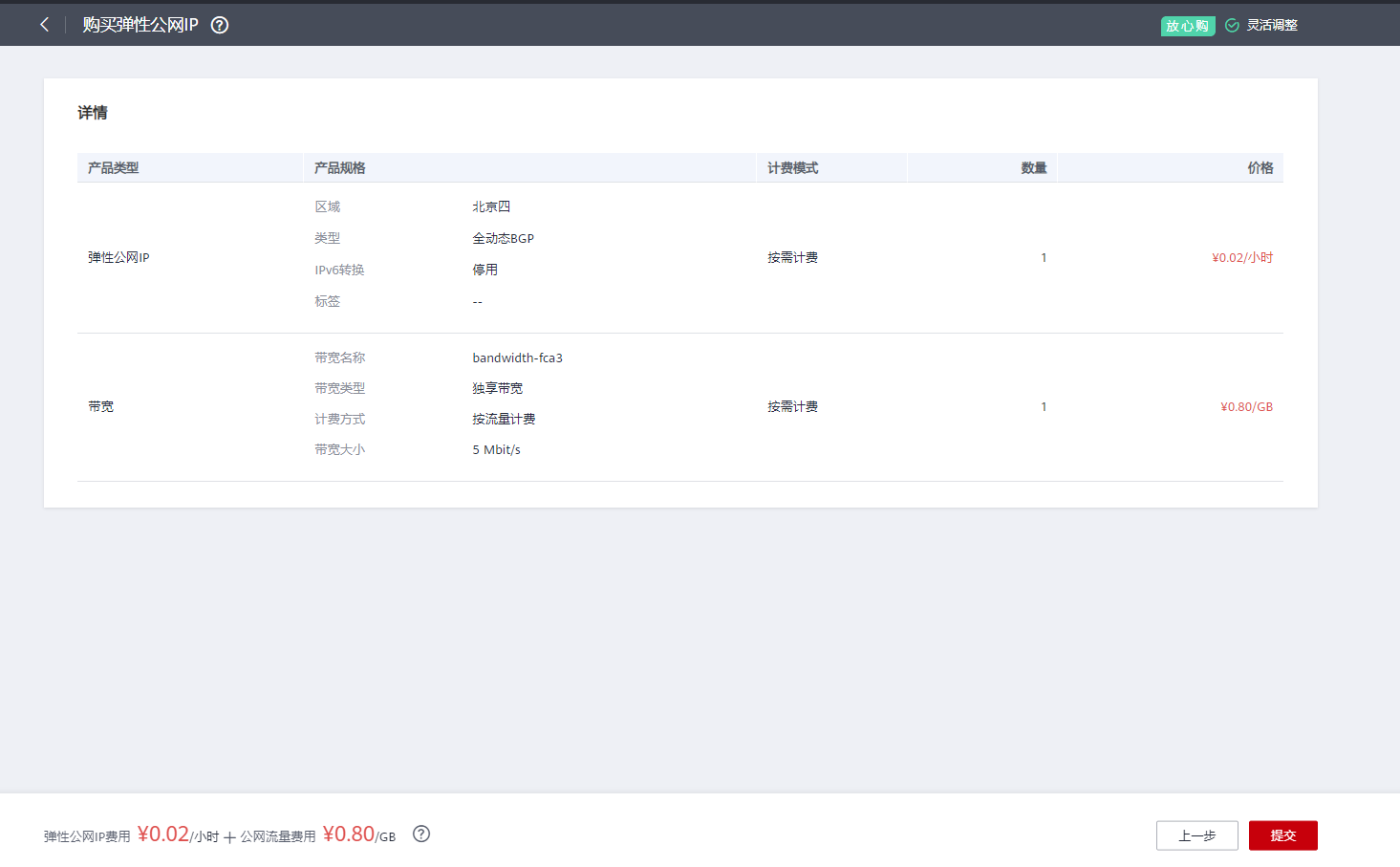
如果没有可用的弹性公网IP，可以单击“查看弹性公网IP“，进入如下页面：



单击购买弹性公网IP，进入如下页面：



计费模式选择按需计费，区域选择与数据库同一个区域（如：华北-北京四），公网带宽选择按流量计费，其它保持不变，然后单击“立即购买“。



确认没问题后，单击提交。



弹性公网IP购买成功，然后回到前面的数据库实例基本信息页面。



单击刷新。



选择弹性公网IP后，单击“确定”。



弹性公网IP绑定成功。

找到内网安全组，单击进入安全组进行8000端口设置，具体如下：



选择入方向规则，然后添加规则，具体如下：



填写好端口8000后，单击“确定“。



8000端口放开成功。

# 使用python连接数据库

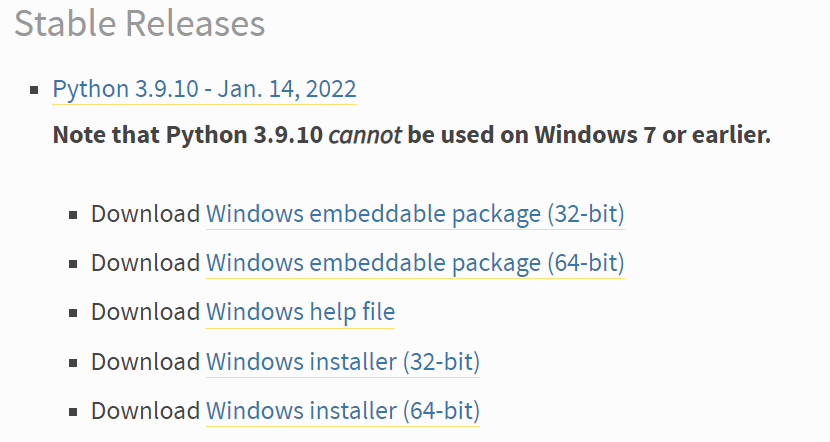
## 下载并安装Python

此实验需要执行连接的计算机上安装Python，如果计算机上已安装Python，可以跳过此节。

下载Python解释器。

在如下链接中下载Python解释器：<https://www.python.org/downloads/windows/>

根据自己电脑的实际选择安装包。

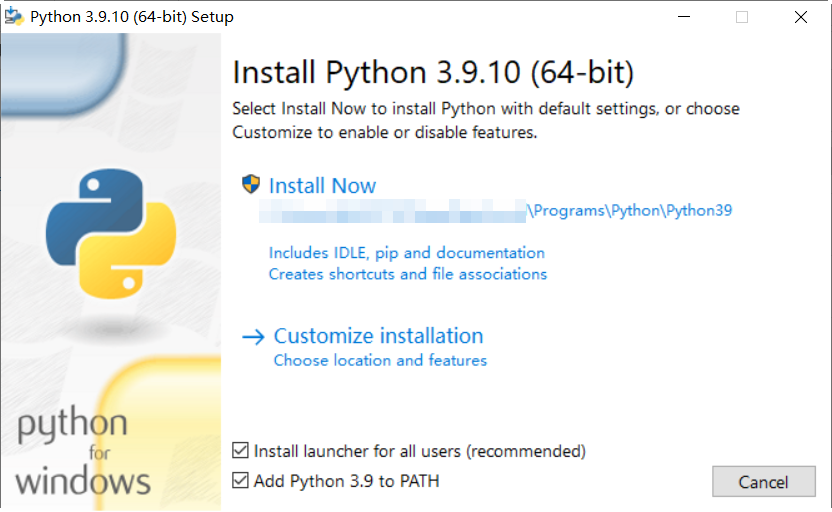


安装Python解释器

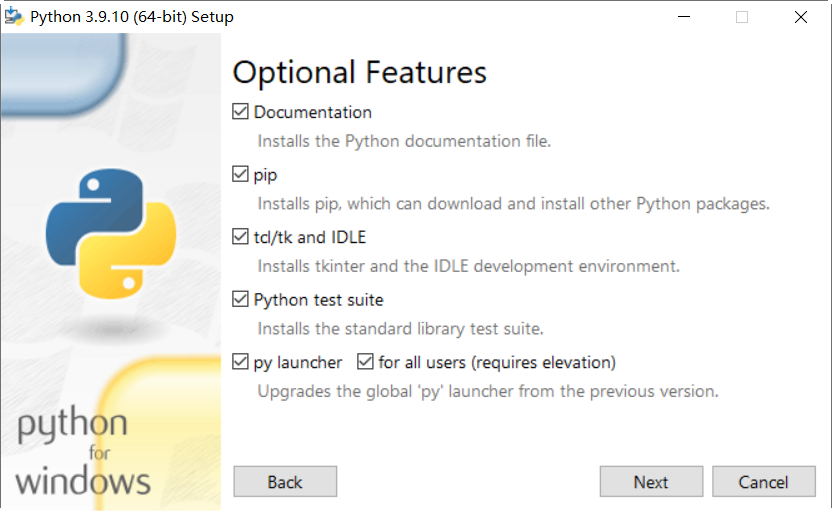
单击安装文件进行安装。



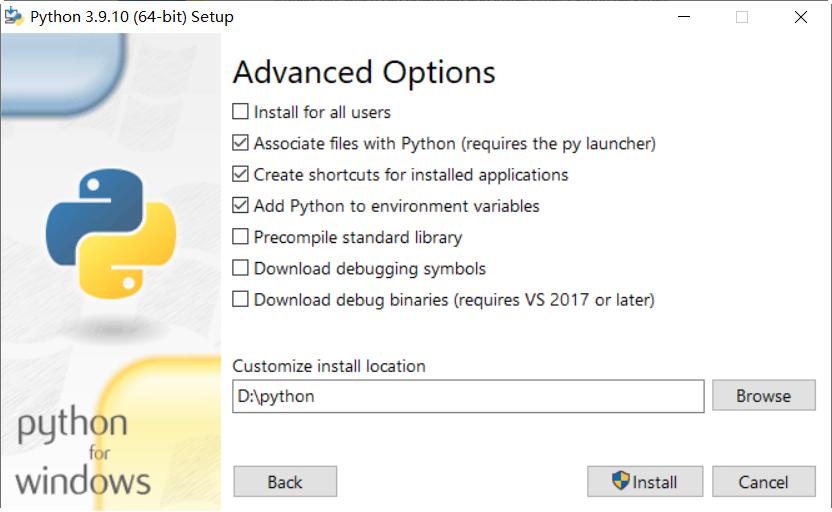
勾选“Add Python 3.9 to PATH”，单击“Customize installation”。



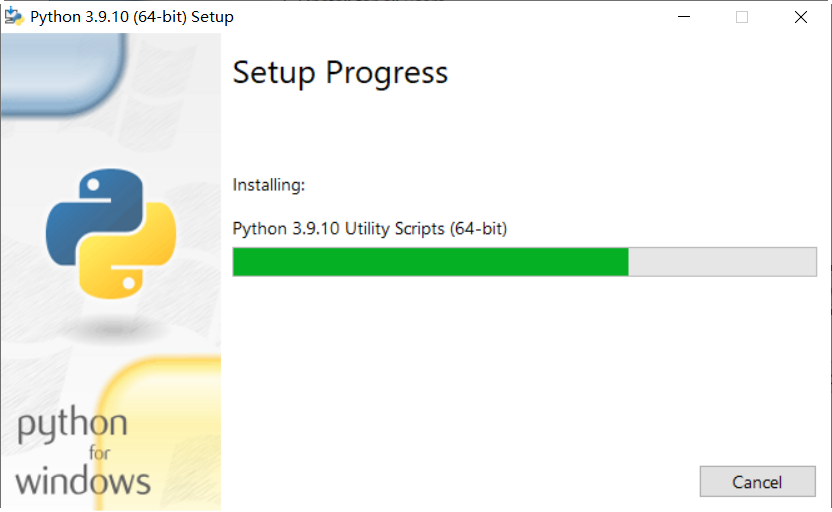
单击“Next”。



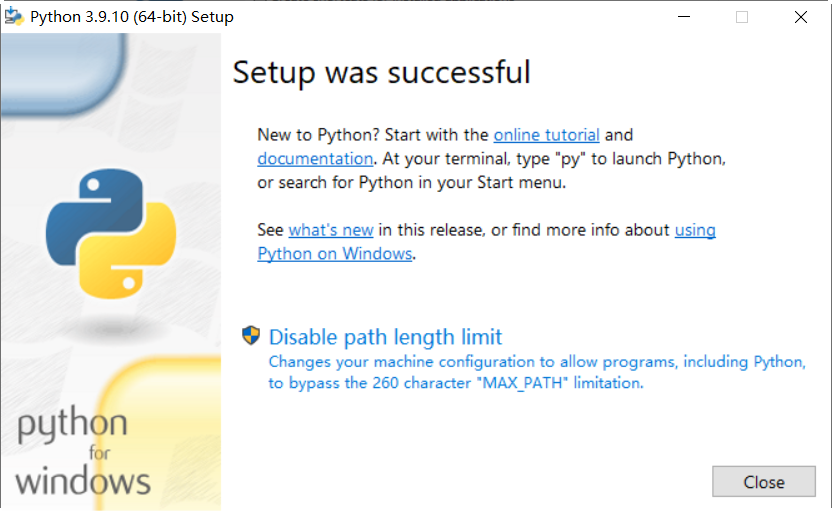
选择安装路径，单击“Install”。



等待安装完成。

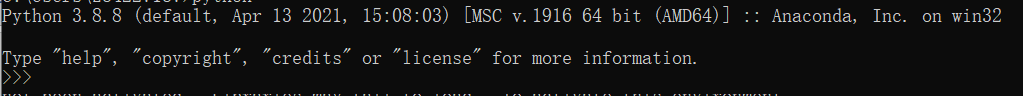


安装成功后，单击Close。



测试是否安装成功

打开 CMD 命令行，输入 Python，进入 Python 环境则安装成功。



## 连接准备

Psycopg是一种用于执行SQL语句的PythonAPI，可以为PostgreSQL、GaussDB数据库提供统一访问接口，应用程序可基于它进行数据操作。

安装通用psycopg2包。

到有访问权限的python镜像服务器中下载并安装psycopg2，此实验中，使用清华的镜像。

命令如下：

pip install --trusted-host pypi.tuna.tsinghua.edu.cn -i https://pypi.tuna.tsinghua.edu.cn/simple/ -U psycopg2

过程和返回结果如下。

Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple/

Collecting psycopg2

Downloading https://pypi.tuna.tsinghua.edu.cn/packages/57/0f/b17f51d21ece594452a70c50d5c153dd25aab182c88497f493b5afbf5820/psycopg2-2.9.3-cp38-cp38-win\_amd64.whl (1.1 MB)

|████████████████████████████████| 1.1 MB 6.8 MB/s

Installing collected packages: psycopg2

Successfully installed psycopg2-2.9.3

通过工单或者联系华为支持工程师开通的Windows下的远程Python连接。

由于GaussDB(for openGauss)对通信协议进行了安全加固，默认采用SHA256的加密方式，这导致与PostgreSQL的默认通信协议MD5互相不兼容，因此，使用psycpog2 的PostgreSQL原生版本默认是不能连接GaussDB(for openGauss)的。会报类似下述错误：

psycopg2.OperationalError: connection to server at "\*\*\*.\*\*\*.\*\*\*.\*\*", port 8000 failed: none of the server's SASL authentication mechanisms are supported

但是，GaussDB(for openGauss)并没有删除MD5的加密和验证逻辑，因此，是可以通过提交工单或联系华为支持工程师修改加密方式，使其支持MD5用于Python教学。

变更了通信协议加密方式后，创建新用户，让Python解释器使用该用户连接数据库，或者重置下已有的数据库帐号密码。

数据库中是不会记录任何形式的用户密码的，只有密码的hash值，在密码校验时使用的是加盐后的hash值比对（也是业界的通用做法）。所以在改变了密码的校验策略后，数据库中并没有存储原密码的MD5值，是仍然无法采用开源兼容的MD5校验的。

使用DAS、gsql等数据库客户端连接数据库后，执行如下语句创建新用户：

CREATE USER python\_user IDENTIFIED BY 'python\_user@123';

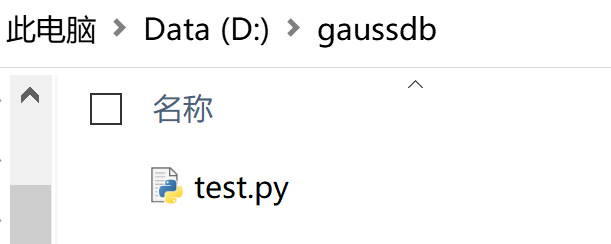
执行结果如下：

NOTICE: The encrypted password contains MD5 ciphertext, which is not secure.

CREATE ROLE

## 连接和使用数据库

新建一个文本文档，将文件命名为test.py，并将其保存到你喜欢的路径下，比如D:\gaussdb。



在test.py文件中加入如下内容。对于斜体部分的内容说明如下：

* user及password的值分别为前面创建的用户及其密码；
* host请输入所要连接的数据库实例公网IP；
* port为实例的端口号，在实例创建时通常默认使用8000，如果做了变更，请使用对应的端口。

import psycopg2

#创建连接对象

conn=psycopg2.connect(database="postgres",user=" python\_user",password=" python\_user@123",host="数据库实例的公网IP",port=8000)

cur=conn.cursor() #创建指针对象

# 创建表

cur.execute("CREATE TABLE student(id integer,name varchar,sex varchar);")

#插入数据

cur.execute("INSERT INTO student(id,name,sex) VALUES(%s,%s,%s)",(1,'Aspirin','M'))

cur.execute("INSERT INTO student(id,name,sex) VALUES(%s,%s,%s)",(2,'Taxol','F'))

cur.execute("INSERT INTO student(id,name,sex) VALUES(%s,%s,%s)",(3,'Dixheral','M'))

# 获取结果

cur.execute('SELECT \* FROM student')

results=cur.fetchall()

print (results)

进入test.py所在目录。

cd /d D:\gaussdb

运行test.py脚本

python test.py

返回如下结果表示执行成功。

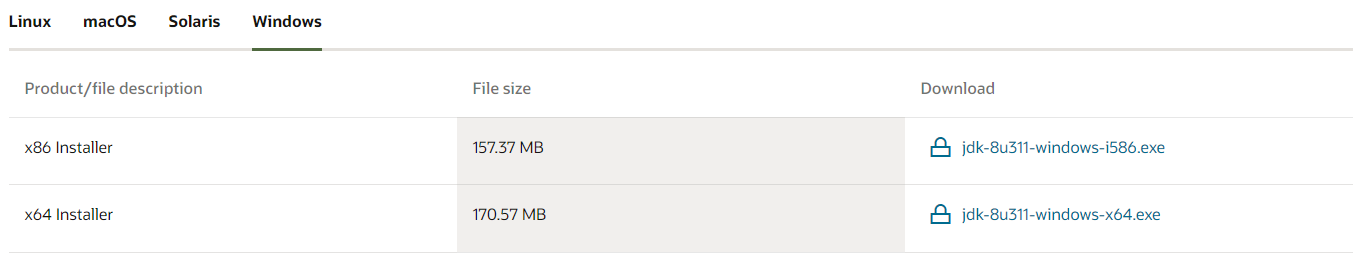
[(1, 'Aspirin', 'M'), (2, 'Taxol', 'F'), (3, 'Dixheral', 'M')]

# 使用JDBC连接数据库

## 下载并安装JDK

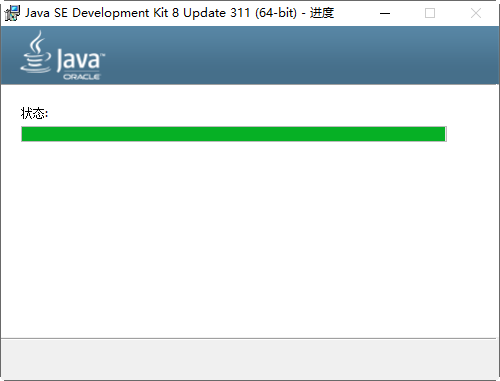
下载JDK，推荐安装java8。根据自己计算机的实际选择安装包

<https://www.oracle.com/java/technologies/downloads/#java8>



双击jdk-8u311-windows-x64.exe进行安装。

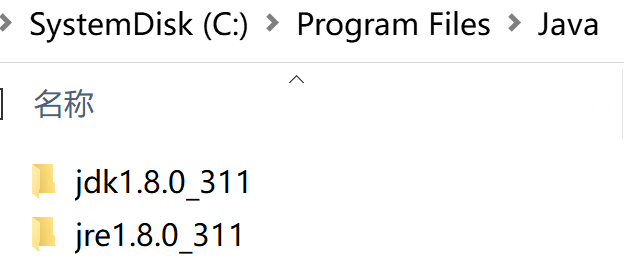
默认设置即可，出现安装进度。



如下显示表示安装成功：



查看安装目录。



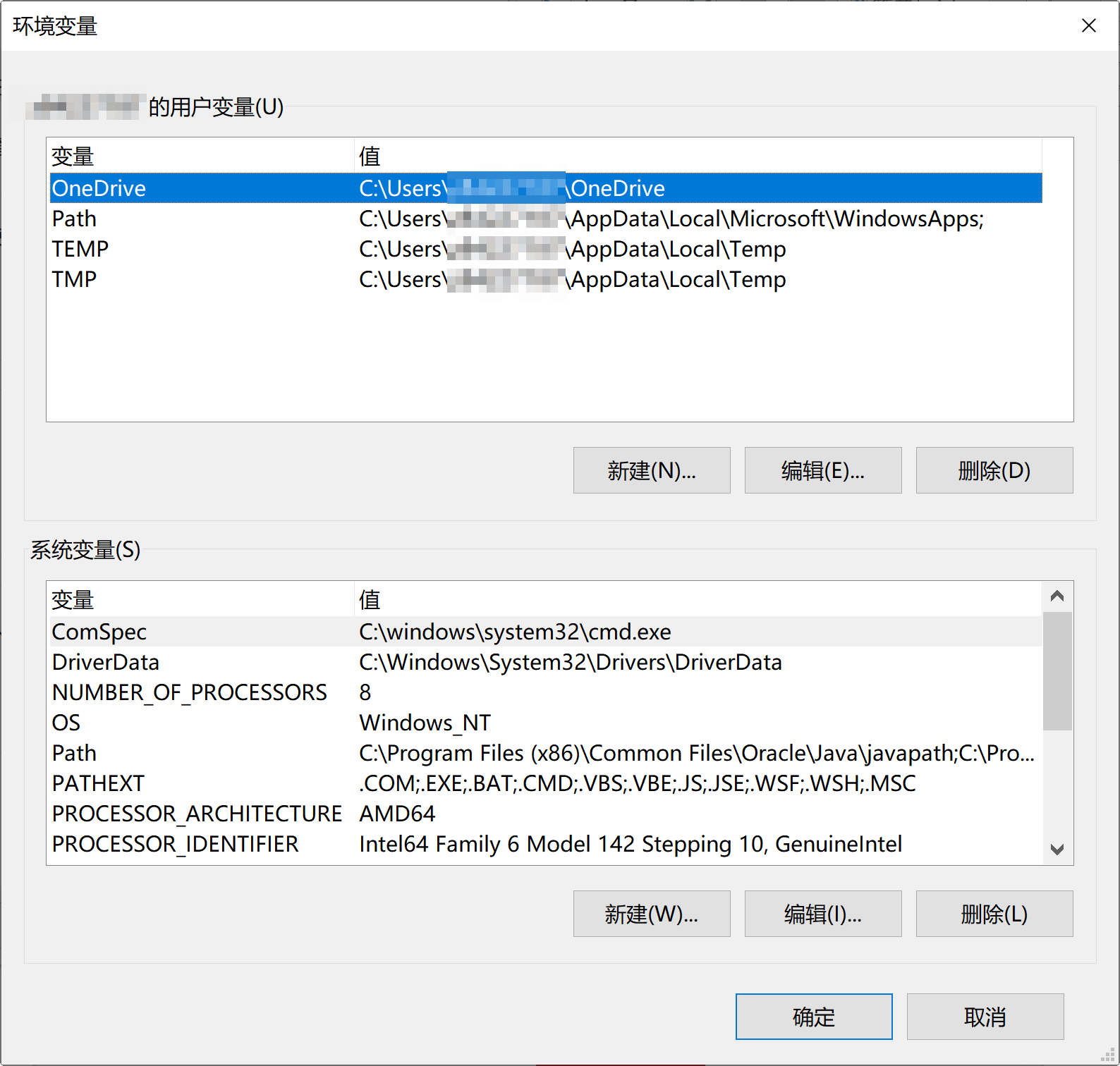
## 配置JDK环境变量

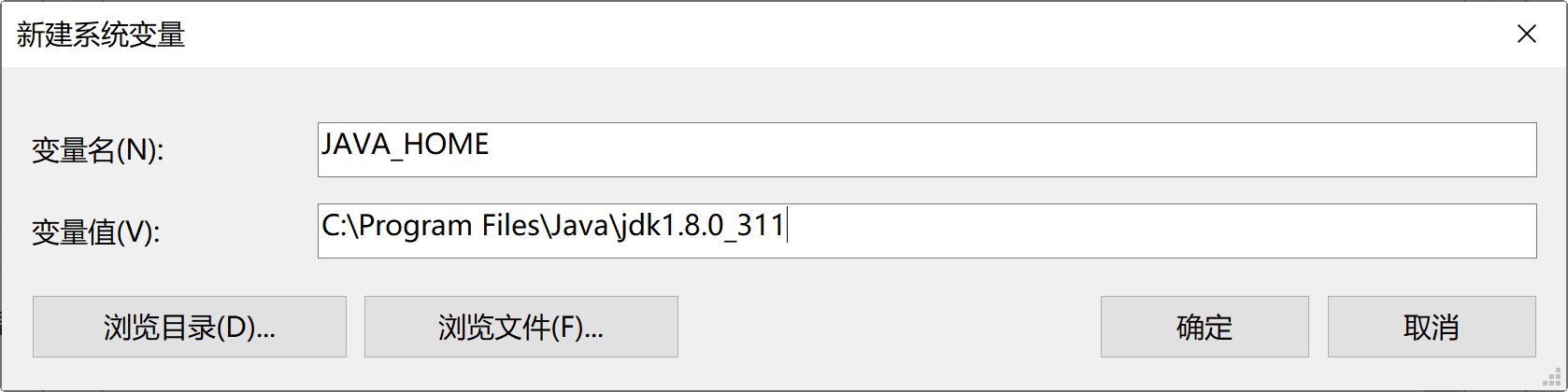
右击“此电脑”选择“属性”，单击“高级系统设置”。





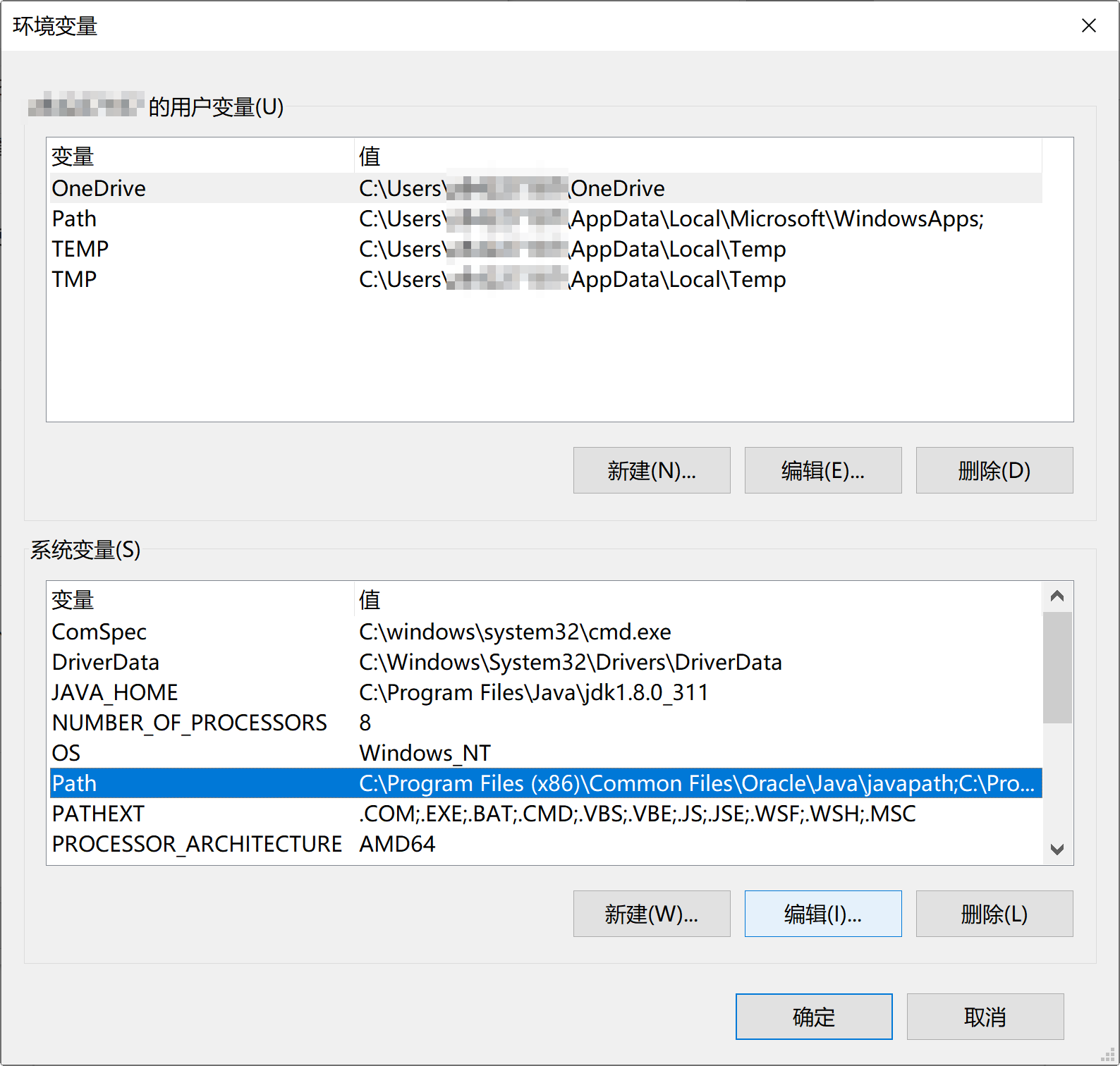
单击“环境变量”，新建系统变量“JAVA\_HOME”，输入JDK安装目录。



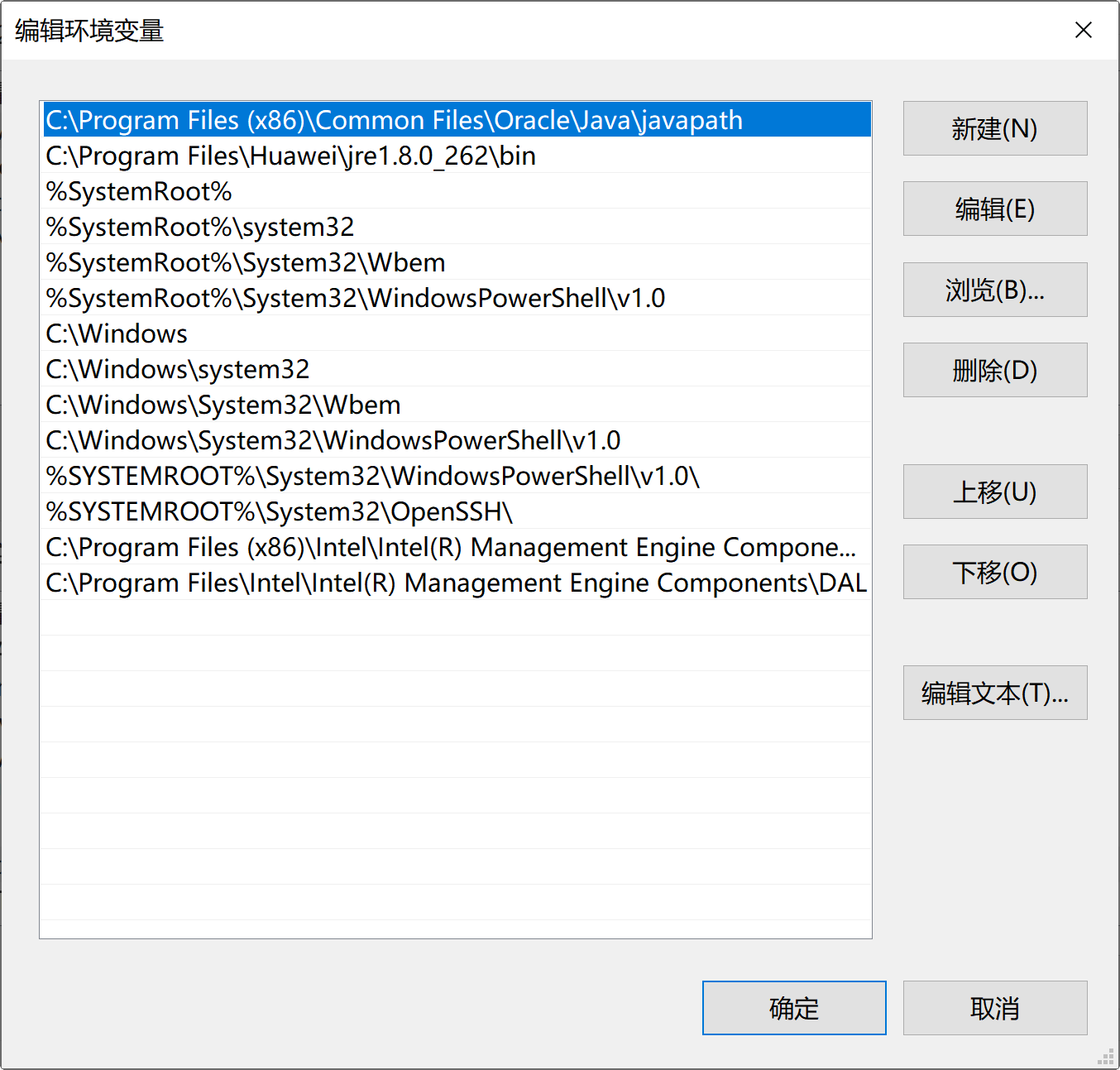


“C:\Program Files\Java\jdk1.8.0\_311 ”为JDK安装目录。

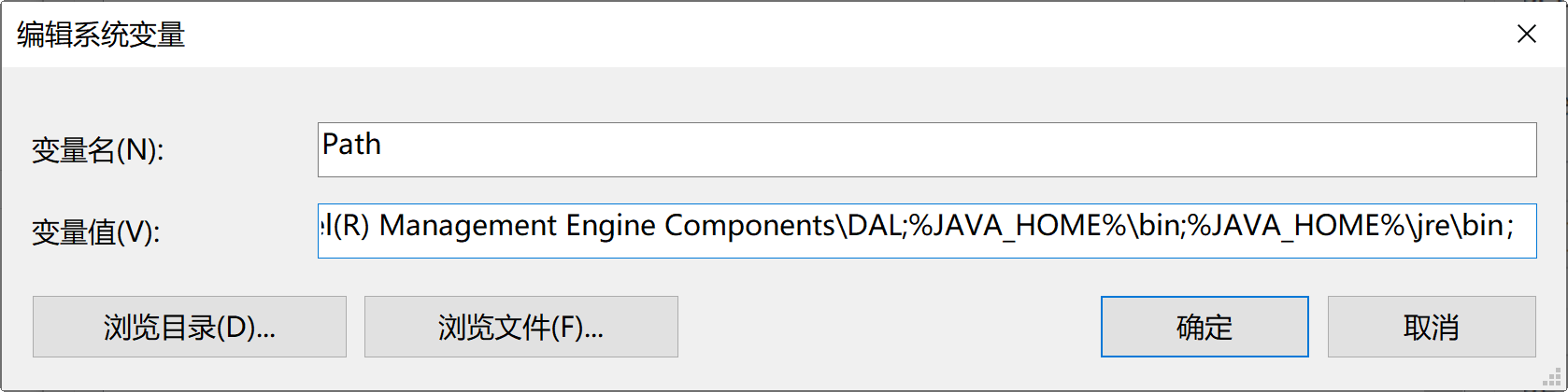
编辑系统变量“path”。



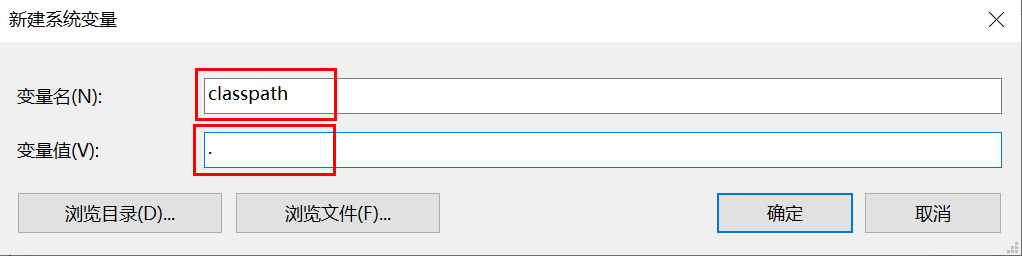
单击“编辑文本”。



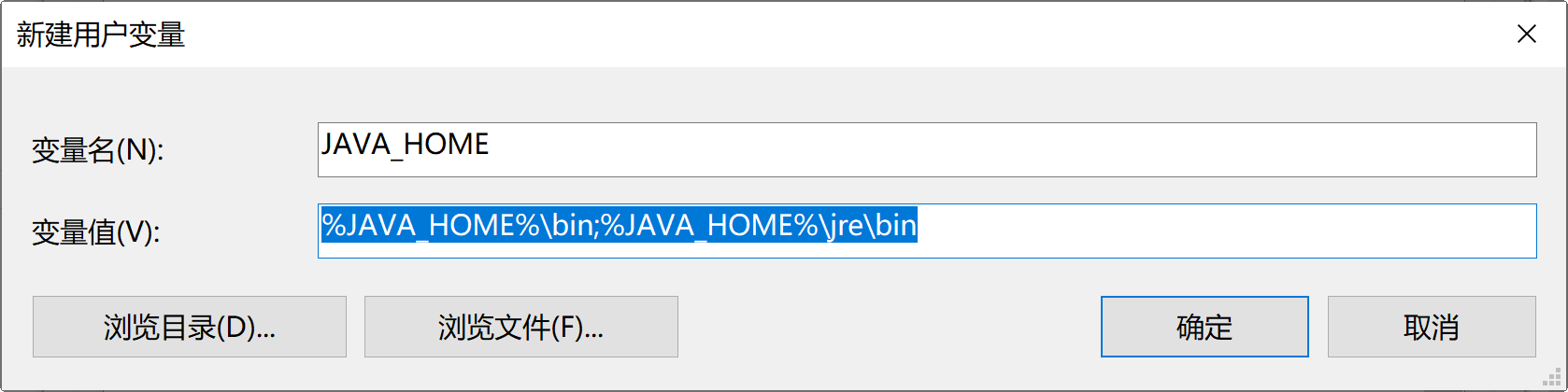
在变量值最后输入： %JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin；**（注意原来Path的变量值末尾有没有;号，如果没有，先输入；号再输入上面的代码）。**



新建系统变量“CLASSPATH”变量，输入“.”即可。



新建用户变量“JAVA\_HOME”，变量值为：%JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin。



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\_311"

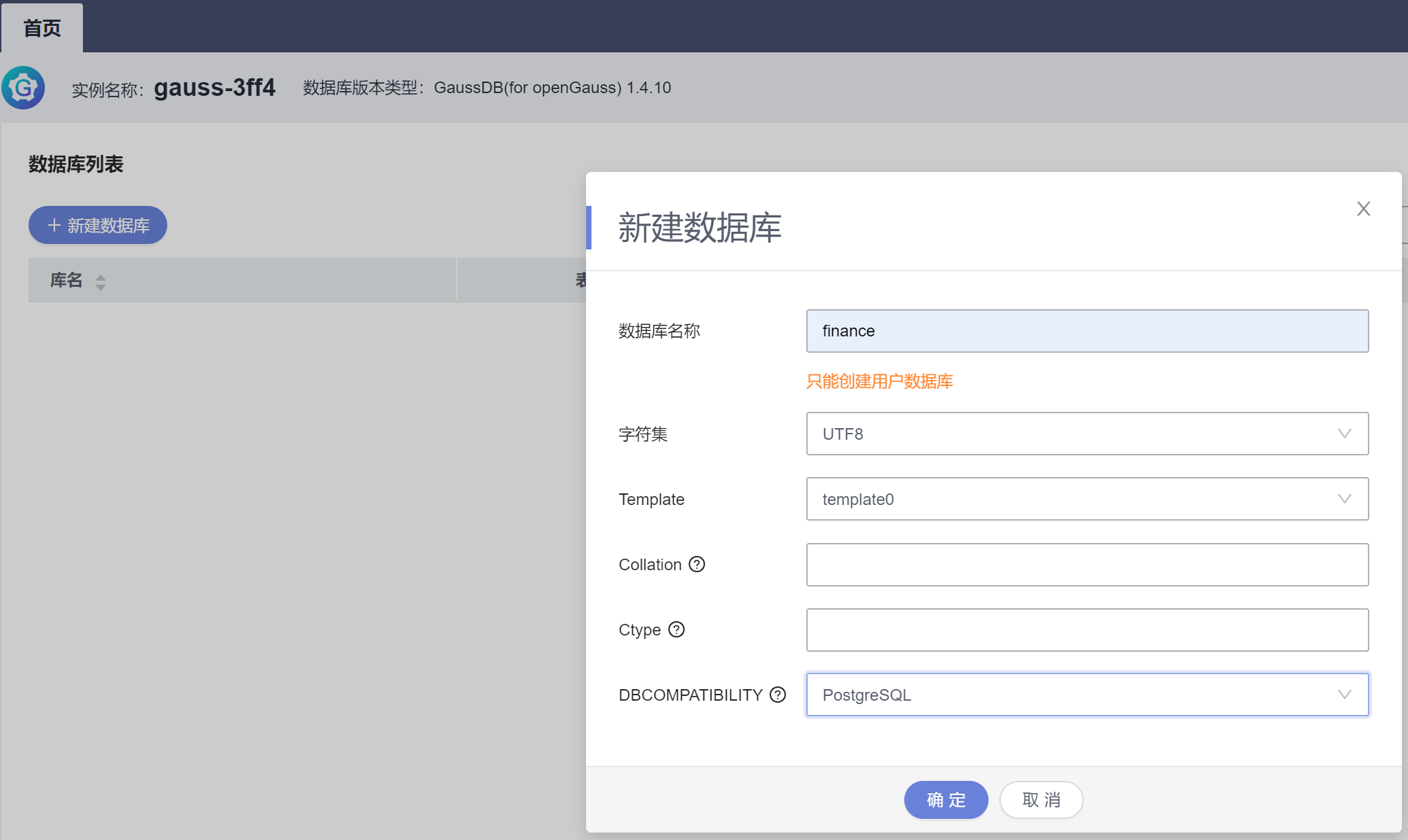
Java(TM) SE Runtime Environment (build 1.8.0\_311-b11)

Java HotSpot(TM) 64-Bit Server VM (build 25.311-b11, mixed mode)

## 连接准备

### 准备实验数据

登录数据库实例后，单击“新建数据库”按钮。



输入新建数据库的数据库名finance，选择对应的数据库字符集（默认使用utf8）后，单击“确定”按钮。

在右侧会提示“创建数据库finance成功”，并在下方的数据库列表中，显示出finance数据库，以上执行完成，表示数据库创建成功。



单击“SQL查询”，进入SQL交互界面。



在finance数据库下创建root用户同名的Schema。

将SQL文本帖入文本框中，单击“执行SQL”按钮或者使用F8执行SQL，在“消息”界面将展示执行结果。

CREATE SCHEMA root;



root Schema创建成功后，将数据库切换至对应的Schema，如下：



在finance库下创建测试表websites。



语句如下：

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/');

### 下载GaussDB(for openGauss)的Java驱动包

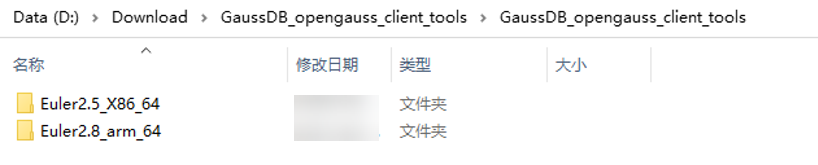
下载连接GaussDB(for openGauss)的Java驱动包。

通过以下链接，下载驱动包

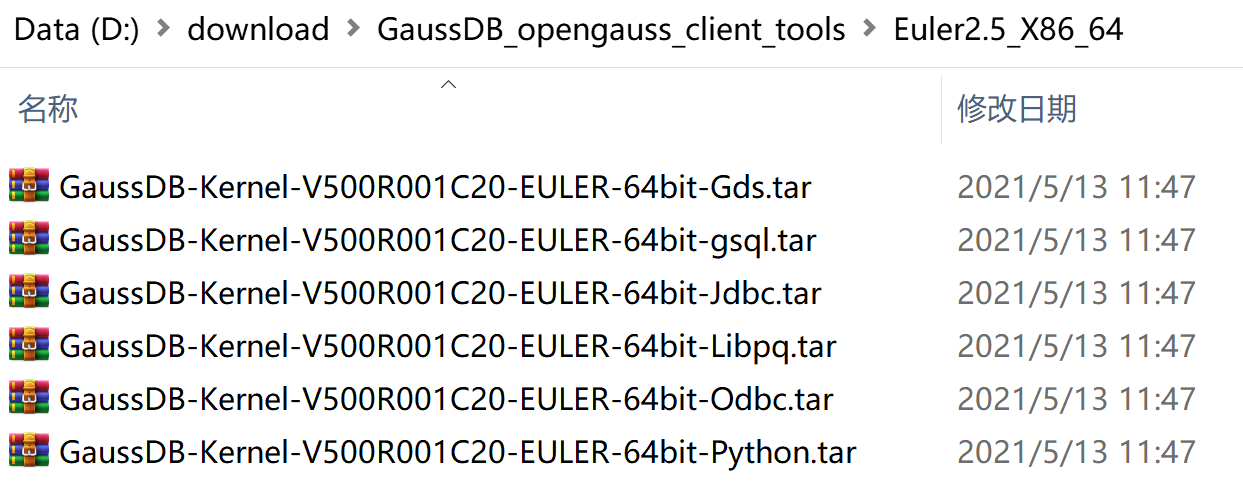
<https://dbs-download.obs.cn-north-1.myhuaweicloud.com/rds/GaussDB_opengauss_client_tools.zip>

假设文件存放在d:\Download目录下，下载的包名为GaussDB\_opengauss\_client\_tools.zip，

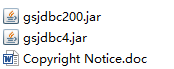
然后对此压缩包进行解压后会有二个文件目录，具体如下：



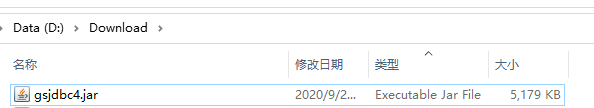
选择与云数据库相应的X86版驱动版本，双击Euler2.5\_X86\_64 进入。



然后再选择GaussDB-Kernel-V500R001C20-EULER-64bit-Jdbc.tar.gz包进行解压，解压后内容如下：



最后将gsjdbc4.jar复制到d:\Download目录下。



## 连接GaussDB(for openGauss)并执行Java代码

使用Java程序连接数据库并进行查询。

在d:\Download\中创建GaussDBopenGaussDemo.java文件，拷贝如下内容到文中，注意红色斜体部分要按照实际情况进行替换。

import java.sql.\*;

public class GaussDBopenGaussDemo{

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

static final String DB\_URL = "jdbc:postgresql://***弹性公网IP*:**8000/***finance***?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 = "***XXX*"**;

static final String PASS = "***XXXXXXXX***";

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 root.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）

cd /d D:\download

D:\Download> javac -encoding utf-8 -cp d:\Download\gsjdbc4.jar GaussDBopenGaussDemo.java

再执行以下命令

D:\Download> java -cp .;D:/Download/gsjdbc4.jar GaussDBopenGaussDemo

执行结果。

执行结果如下：

连接数据库...

十一月 18, 2021 3:02:54 下午 org.postgresql.core.v3.ConnectionFactoryImpl openConnectionImpl

信息: [3977357f-a502-46c8-8596-fda26bc89a69] Try to connect. IP: 123.60.214.38:8000

十一月 18, 2021 3:02:54 下午 org.postgresql.core.v3.ConnectionFactoryImpl openConnectionImpl

信息: [7.250.106.229:50549/123.60.214.38:8000] Connection is established. ID: 3977357f-a502-46c8-8596-fda26bc89a69

十一月 18, 2021 3:02:54 下午 org.postgresql.core.v3.ConnectionFactoryImpl openConnectionImpl

信息: Connect complete. ID: 3977357f-a502-46c8-8596-fda26bc89a69

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

## 加载GaussDB(for openGauss) ODBC驱动包

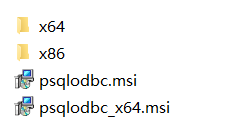
下载GaussDB(for openGauss) Windows版ODBC驱动包。

从如下地址下载GaussDB\_opengauss\_client\_tools.zip。

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

解压缩发布包，从中获取包名为GaussDB-Kernel-V500R001C10-Windows-Odbc.tar.gz的压缩包。

解压GaussDB-Kernel-V500R001C10-Windows-Odbc.tar.gz，得到如下文件和文件夹。



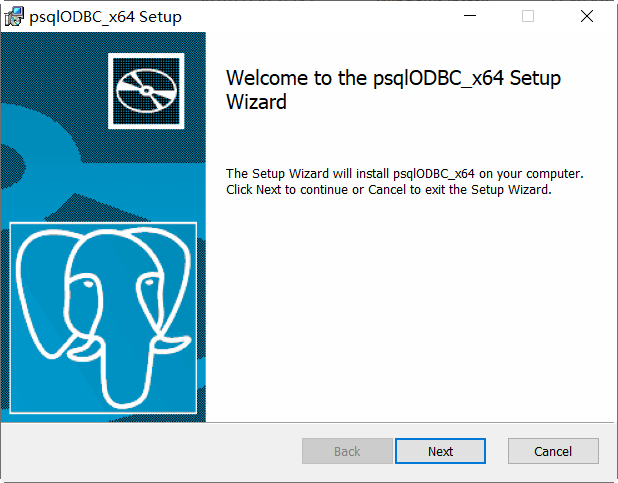
根据需要，双击psqlodbc.msi（32位）或者psqlodbc\_x64.msi（64 位）进行驱动安装。

64位操作系统上进行32位程序开发，安装32位驱动程序。

64位操作系统上进行64位程序开发，安装64位驱动程序。

本实验以64位驱动的安装和配置为例进行说明，32位的安装配置相似。

单击psqlodbc\_x64.msi后出现如下安装向导，根据安装向导的指引进行安装即可。



## 配置数据源

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。

打开ODBC数据源管理器。

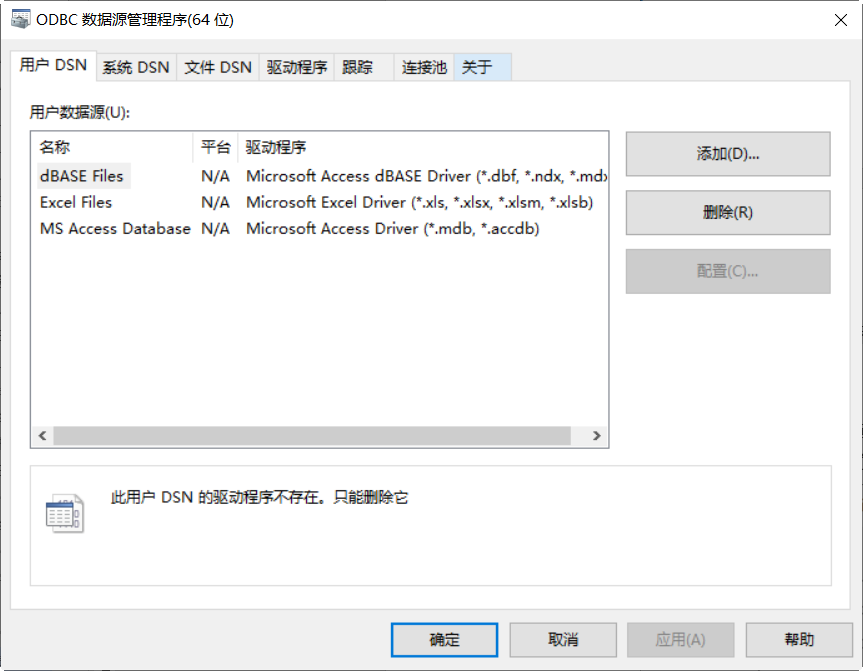
在Windows左下角的搜索框中输入ODBC，即可出现数据源管理器。选择ODBC数据源（64位）。



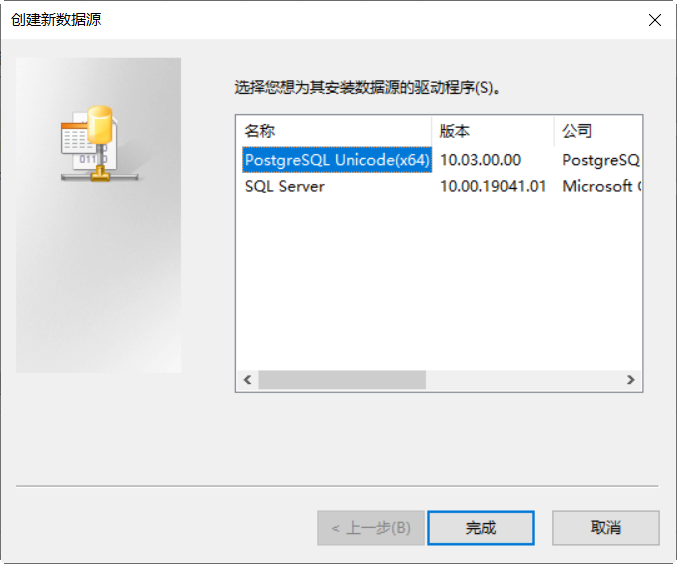
**说明：**如果要进行32位的程序开发，请选择ODBC Data Sources(32-bit)。

配置数据源。

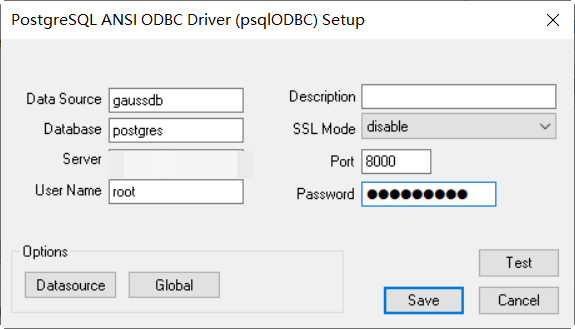
在打开的驱动管理器上，在“用户DSN”页签，单击“ 添加 ”。



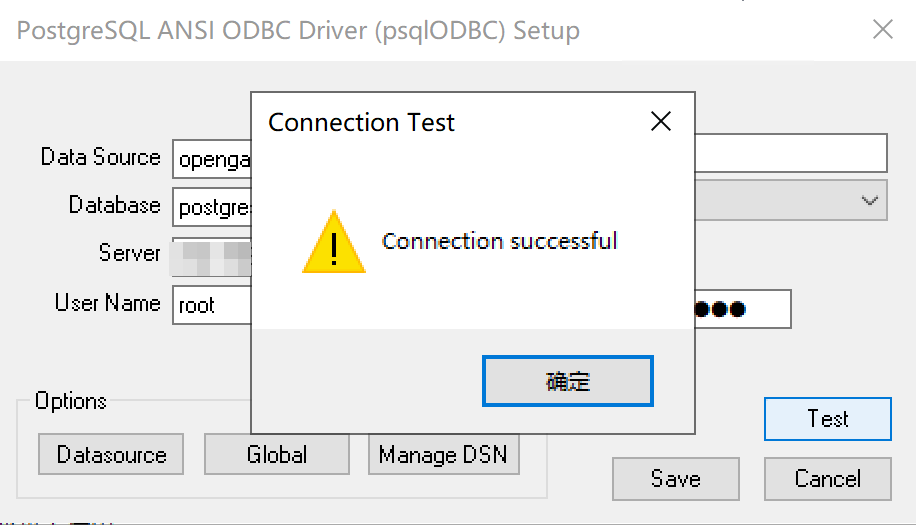
在“创建新数据源”对话框中选择” PostgreSQL Unicode(x64)”。



在“PostgreSQL ANSI ODBC Driver (psqlODBC) Setup”对话框中进行配置。

* **Data Source：**根据喜好设置数据源名称，例如gaussdb。
* **Database：**设置为需要连接的数据库名称，例如此处连接的是GaussDB(for openGauss)实例创建后的初始数据库postgres。
* **Sever：**请输入所要连接的数据库实例的IP地址，如果是外网连接GaussDB(for openGauss)，请输入实例所绑定的弹性公网IP地址。
* **Port：**请输入购买数据库实例时设置的端口，通常为8000。
* **User Name/Password：**请输入连接数据库的用户名和密码。例如购买数据库实例时所创建的root用户及其密码。

输入完成后，单击“Test”，测试是否可以成功连接数据库。弹出“Connection successful”，表示可以成功连接数据库。



## 执行示例代码连接并操作数据库

将如下示例代码拷贝到C/C++ IDE中。

**注意：**红色字体请根据实际替换上一节所设置的数据源名称。

// 此示例演示如何通过ODBC方式获取GaussDB(for openGauss)中的数据。

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

#include <windows.h>

#include <stdlib.h>

#include <stdio.h>

#include <sqlext.h>

#ifdef WIN32

#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 = 100;

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“userName”与“password”分别表示连接数据库的用户名和用户密码，请根据实际情况修改。

// 如果odbc.ini文件中已经配置了用户名密码，那么这里可以留空（""）；但是不建议这么做，因为一旦odbc.ini权限管理不善，将导致数据库用户密码泄露。

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

(SQLCHAR\*) "", SQL\_NTS, (SQLCHAR\*) "", 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,100,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);

}

编译运行上述示例代码，返回如下内容，表示示例代码执行成功。

Connected !

SQLColAtrribute

SQLGetData ----ID = 25

SQLGetData ----ID = 100

Done !

--------------------------------

Process exited after 0.6748 seconds with return value 0

此次试验结束。