《数据库》

openGauss二次开发指导手册



华为技术有限公司

目录

[前 言 3](#_Toc80620140)

[简介 3](#_Toc80620141)

[内容描述 3](#_Toc80620142)

[前置条件 3](#_Toc80620143)

[实验环境说明 3](#_Toc80620144)

[实验概览 4](#_Toc80620145)

[1 社区环境准备 5](#_Toc80620146)

[1.1 注册码云账号 5](#_Toc80620147)

[1.2 绑定主邮箱 6](#_Toc80620148)

[1.3 签署CLA贡献者协议 6](#_Toc80620149)

[2 搭建开发环境 8](#_Toc80620150)

[2.1 拉取个人分支 8](#_Toc80620151)

[2.2 安装并配置Git环境 9](#_Toc80620152)

[2.2.1 安装Git环境 9](#_Toc80620153)

[2.2.2 配置Git环境 10](#_Toc80620154)

[2.3 安装开发工具 12](#_Toc80620155)

[3 二次开发（以实现gauss\_hello函数为例） 15](#_Toc80620156)

[3.1 将个人远程仓库内容更新至本地 15](#_Toc80620157)

[3.2 修改本地文件 15](#_Toc80620158)

[3.3 提交本地修改到个人远程仓库 19](#_Toc80620159)

[4 编译 20](#_Toc80620160)

[4.1 编译环境要求 20](#_Toc80620161)

[4.1.1 操作系统环境要求 20](#_Toc80620162)

[4.1.2 软件依赖要求 20](#_Toc80620163)

[4.2 编译环境准备（使用代金券购买华为云ECS服务） 21](#_Toc80620164)

[4.2.1 登录华为云 21](#_Toc80620165)

[4.2.2 购买弹性云服务器ECS 22](#_Toc80620166)

[4.3 版本编译 26](#_Toc80620167)

[4.3.1 代码下载 26](#_Toc80620168)

[4.3.2 软件安装编译 29](#_Toc80620169)

[5 验证二次开发结果 33](#_Toc80620170)

[5.1 安装编译后的openGauss安装包 33](#_Toc80620171)

[5.1.1 修改操作系统配置 33](#_Toc80620172)

[5.1.2 规划数据库安装包目录 34](#_Toc80620173)

[5.1.3 创建XML配置文件 35](#_Toc80620174)

[5.1.4 初始化安装环境 37](#_Toc80620175)

[5.1.5 执行安装 39](#_Toc80620176)

[5.1.6 安装生成的目录 40](#_Toc80620177)

[5.2 登录数据库验证结果 42](#_Toc80620178)

[附录一：Linux操作系统相关命令 43](#_Toc80620179)

# 前 言

## 简介

openGauss是开源关系型数据库，采用客户端/服务器，单进程多线程架构，支持单机和一主多备部署方式，备机可读，支持双机高可用和读扩展。

本实验主要描述如何对openGauss开源数据库进行二次开发，并编译验证。

## 内容描述

本实验主要内容为在开源社区注册，拉取本地代码，对openGauss开源数据库进行简单修改体验二次开发，并安装数据库进行验证。

## 前置条件

由于本实验涉及到在openEuler操作系统上进行openGauss数据库的二次开发，因此需要掌握Linux系统的基本操作和系统命令，详细请参见[**附录一**](#_附录一：Linux操作系统相关命令_1)。

## 实验环境说明

组网说明

本实验环境需要本地Windows操作系统的PC以及openEuler操作系统的华为云环境，需要购买弹性云服务器。

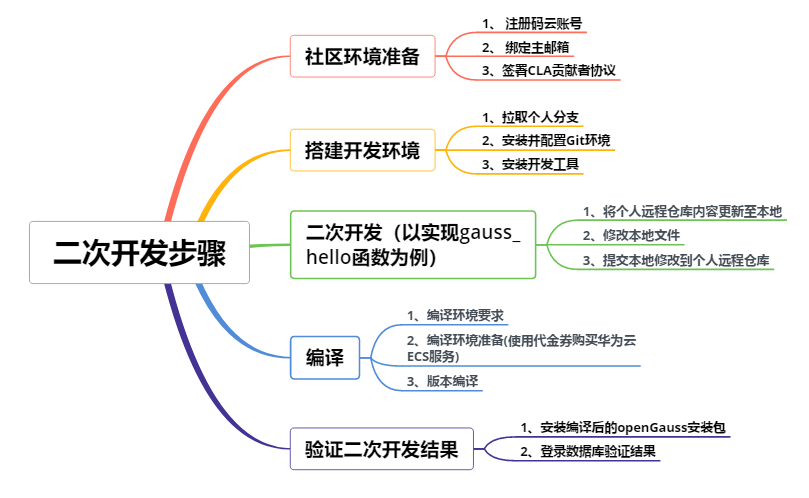
设备介绍

为了满足实验需要，建议每套实验环境采用以下配置：

配置明细表

|  |  |
| --- | --- |
| 软件类型 | 软件版本 |
| 开发环境： | |
| 操作系统 | Windows |
| 编译环境： | |
| 操作系统 | openEuler 20.3 LTS |
| Python | Python 3.7.X |

## 实验概览

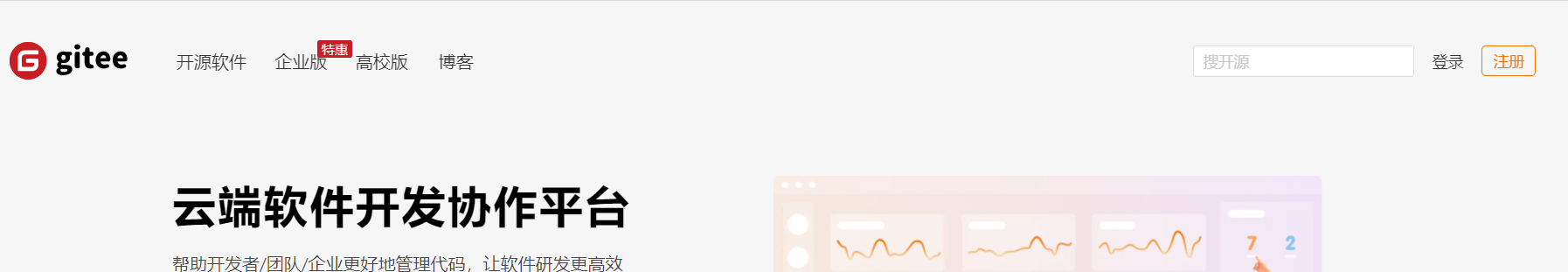


# 社区环境准备

需要完成码云账号注册、绑定主邮箱以及签署CLA贡献者协议之后，才能参与社区贡献。

## 注册码云账号

登录码云官网（<https://gitee.com/>），单击页面右上角的＂注册＂按钮注册账号。



在注册页面填写相关信息并勾选阅读并同意相关条款及规范，完成码云注册，如下图所示。



必填信息如下：

* 姓名：码云账号名称。
* 个人空间地址：注册账号主页地址，可用作推拉代码或者登录Gitee的用户名。
* 手机或者邮箱：验证需要。建议使用邮箱注册，默认为主邮箱，无需执行绑定主邮箱操作。
* 密码：密码不少于6位。建议为字母、符号和数字的组合，以提升安全性。

## 绑定主邮箱

在＂个人主页＂页面，单击＂设置＂按钮进入＂个人信息＂设置页面。如下图所示。



在左侧导航栏单击＂邮箱管理＂，修改或者绑定主邮箱。主邮箱默认为注册时使用的邮箱。如下图所示。



## 签署CLA贡献者协议

进入CLA签署页面（<https://opengauss.org/en/cla.html>）。



选择＂签署个人CLA（Sign Individual CLA）＂，填写相关信息。



按照提示填写信息，Gitee-ID是您自己的gitee账号，邮箱也要与gitee账号的邮箱（以及git初始设置的邮箱）保持一致。然后阅读＂隐私政策＂，如果同意就勾选前面的选择框，然后单击签署，完成CLA签署。如下图：



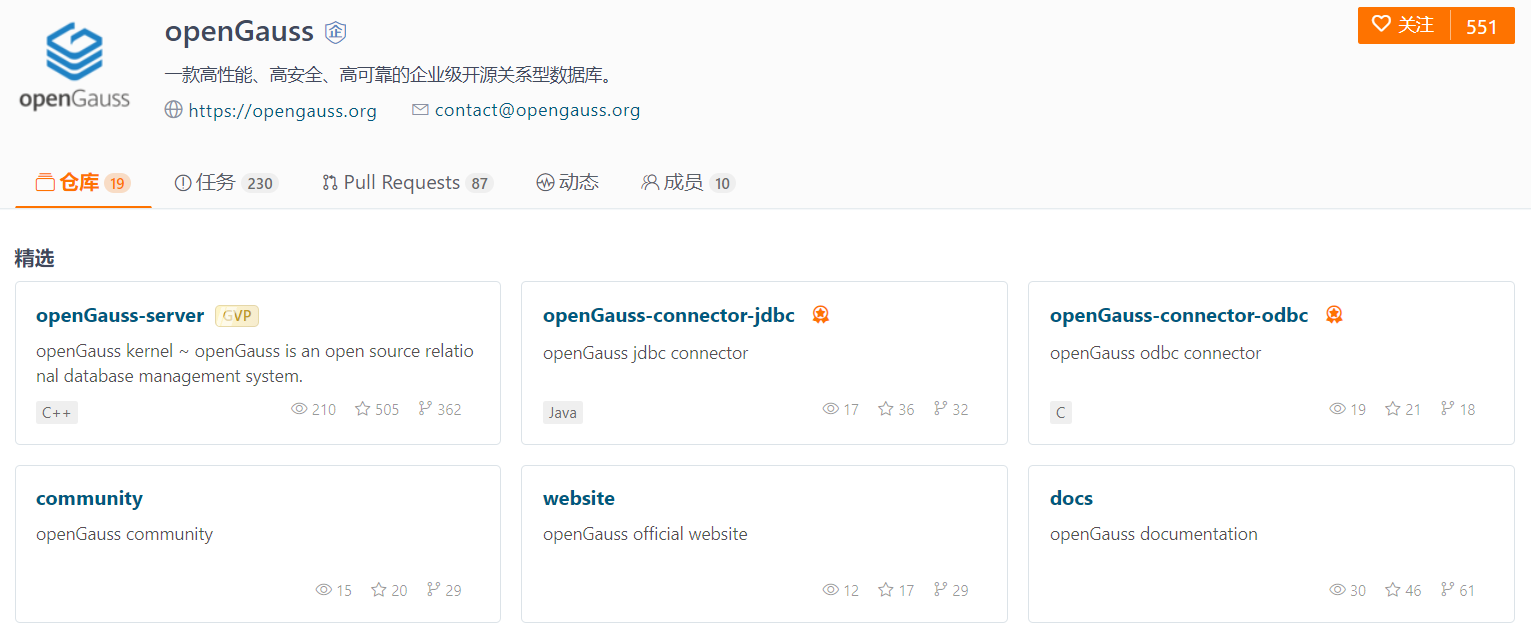
# 搭建开发环境

搭建开发环境之前需要在码云（<https://gitee.com/>）上完成注册个人Gitee账号、签署CLA协议(贡献者许可协议)，详情请参见1 社区环境准备。

## 拉取个人分支

进入开源社区网站的openGauss-server（代码）仓库。

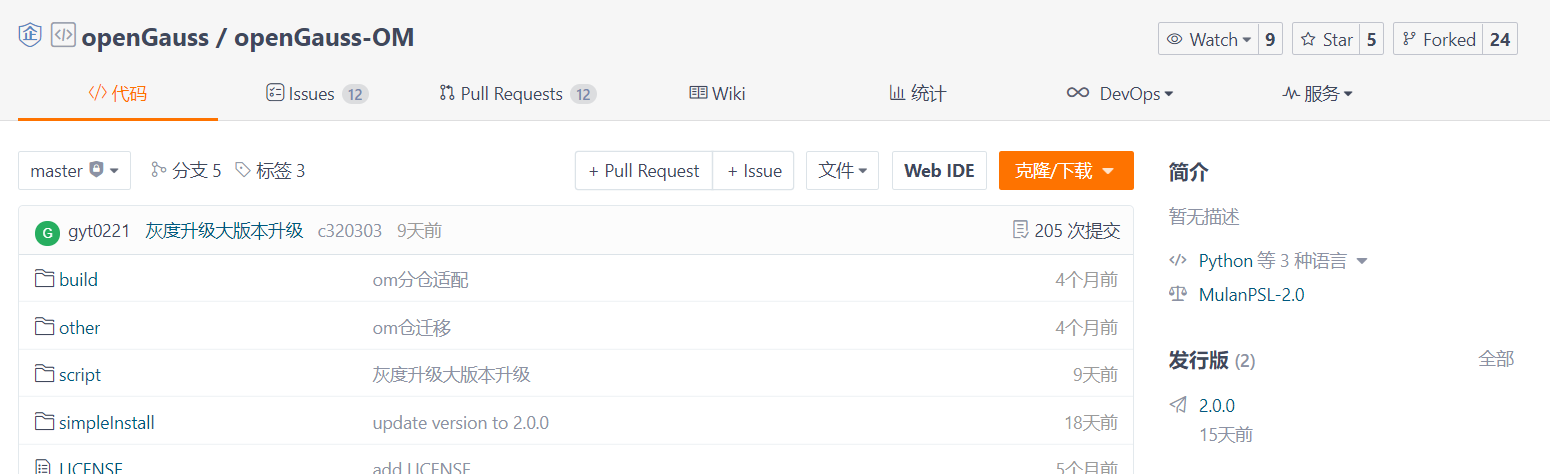
地址如下<https://gitee.com/opengauss>，单击openGauss-server，如下图所示。



单击右上角＂Forked＂按钮，Fork（复刻）个人分支，如下图所示。



进入开源社区网站的openGauss-OM（代码）仓库。

地址如下https://gitee.com/opengauss/openGauss-OM?\_from=gitee\_search，单击右上角＂Forked＂按钮，Fork（复刻）个人分支。如下图所示：

## 安装并配置Git环境

### 安装Git环境

下载Git客户端并按默认设置安装，Git下载地址：<https://git-scm.com/download/win>，根据操作系统位数（32位/64位），下载并安装相应的Git（命令行工具）。

说明：

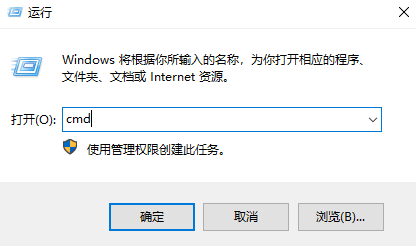
本章节中开发环境使用的是Windows 64位环境，请下载Winodws 64位版本的git工具。



在Windows上下载完成后文件名为Git-2.31.1-64-bit.exe，双击此文件进行安装。

### 配置Git环境

打开运行窗口，并输入cmd，点击＂回车＂进入命令提示符界面。



在cmd中配置全局Git用户名。

git config --global user.name "**Your Name**"

"Your Name"为自己的Gitee账号名称，您可以在Gitee个人主页获取，如下图中xzp-blog。



在cmd中配置全局Git邮箱。

git config --global user.email "**email@example.com**"

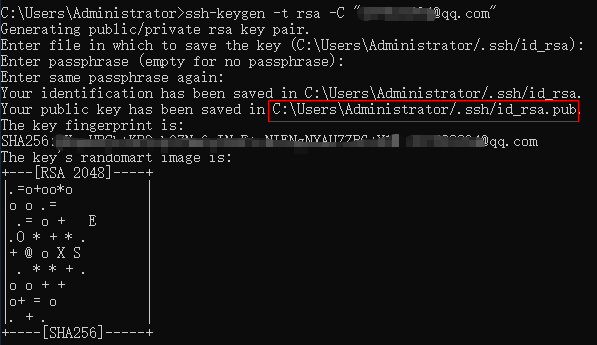
"email@example.com"为注册Gitee的主邮箱，可通过Gitee个人设置中的＂邮箱管理＂获取自己的主邮箱，如下图：



在cmd中生成SSH公钥。

通过Gitee的主邮箱生成SSH公钥。

ssh-keygen -t rsa -C "**email@example.com**"



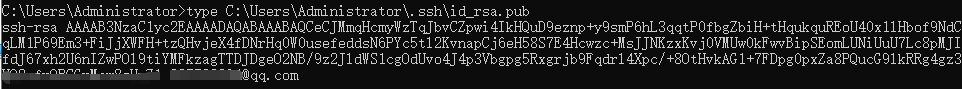
查看公钥信息。

在命令提示符界面中，使用type命令查看公钥信息，公钥路径示例如上图所示。

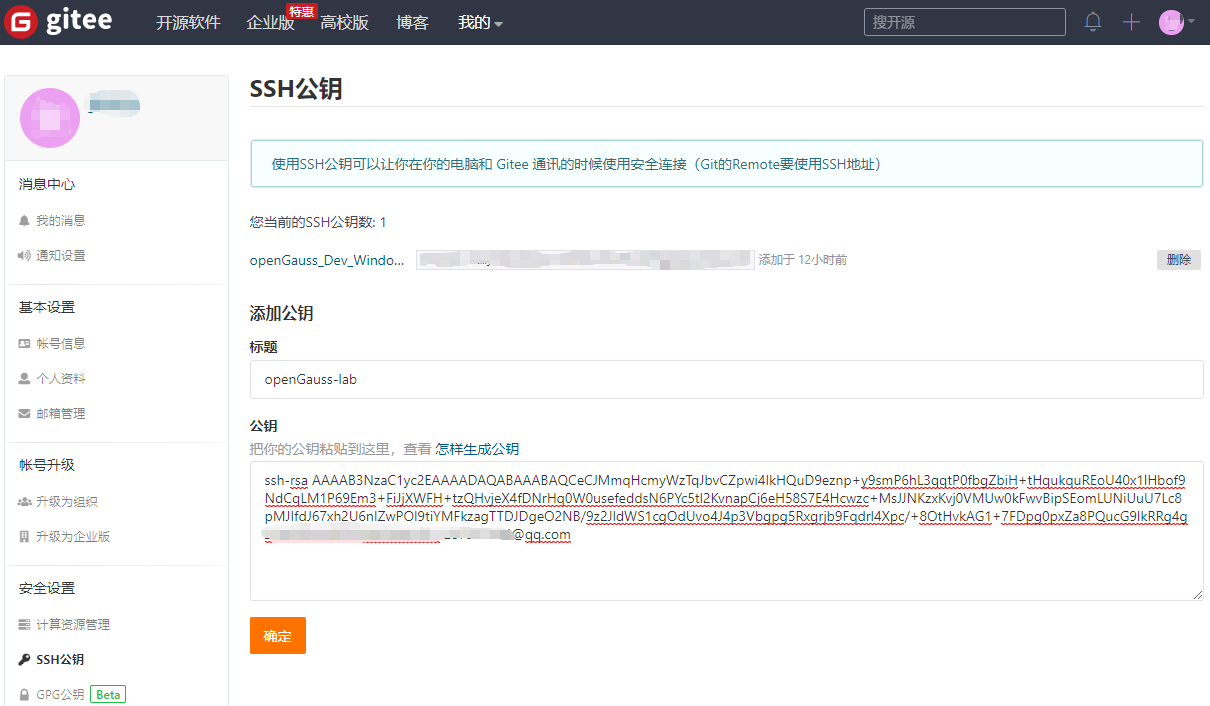
type C:\Users\Administrator\.ssh\id\_rsa.pub

说明

.ssh/id\_rsa.pub为公钥文件保存地址。id\_rsa.pub为自定义的pub文件名称。



在个人Gitee账户中添加SSH公钥。



将远程仓库克隆至本地。

先在本地电脑上创建openGauss-server文件夹，如在C盘下创建。

然后在cmd命令行中，通过如下语句进入存放个人远程仓库的本地目录。

cd C:\openGauss-server

从个人远程仓库地址下载数据库源码至本地工作目录。

git clone [https://gitee.com/**Your Name**/openGauss-server.git](https://gitee.com/Your%20Name/openGauss-server.git)

Your Name为自己配置的全局Git用户名，请根据实际替换。



设置本地工作目录的upstream源。

cd C:\openGauss-server\openGauss-server **---#先进入相应目录**

git remote add upstream <https://gitee.com/opengauss/openGauss-server.git> **---#openGauss远程地址**

设置同步方式。

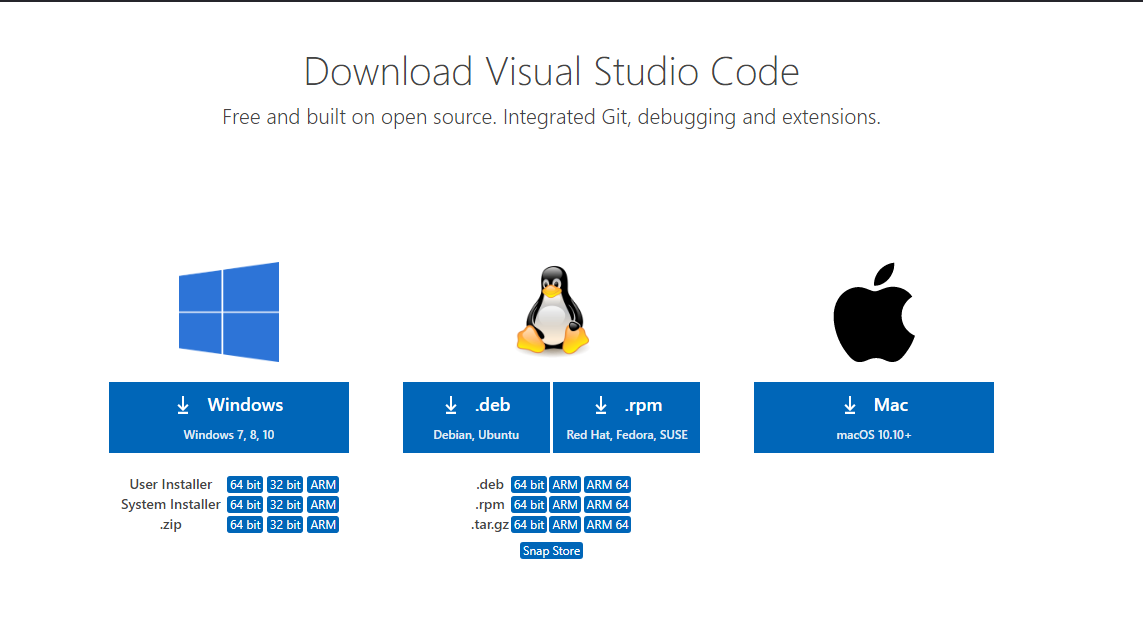
git remote set-url --push upstream no\_push

## 安装开发工具

openGauss内核开发以C/C++语言为主，本节以安装VSCode（Visual Studio Code）工具为例进行介绍。如本地PC已安装有开发工具，可跳过本节。

下载VSCode软件。

登录[https://code.visualstudio.com/download](https://code.visualstudio.com/downloadg) 下载软件如下图所示。请根据操作系统选择相应的版本，本节以64位Windows操作系统为例。



单击可执行文件，对下载的软件进行安装。

选择＂我同意此协议＂后单击＂下一步＂。

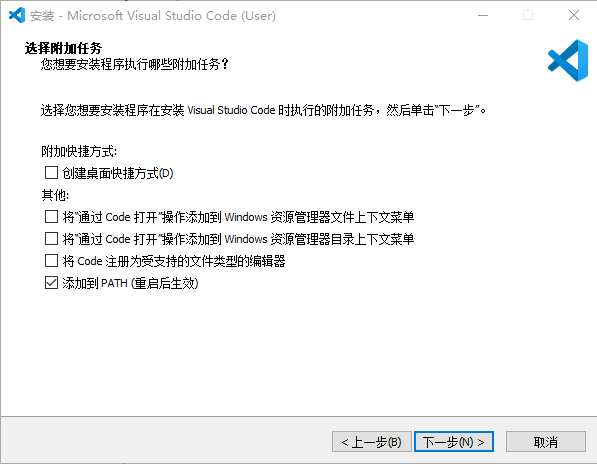


确认安装位置后单击＂下一步＂。

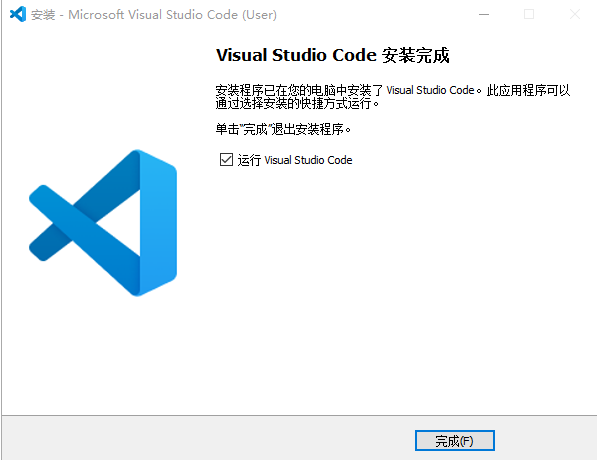


配置环境变量。

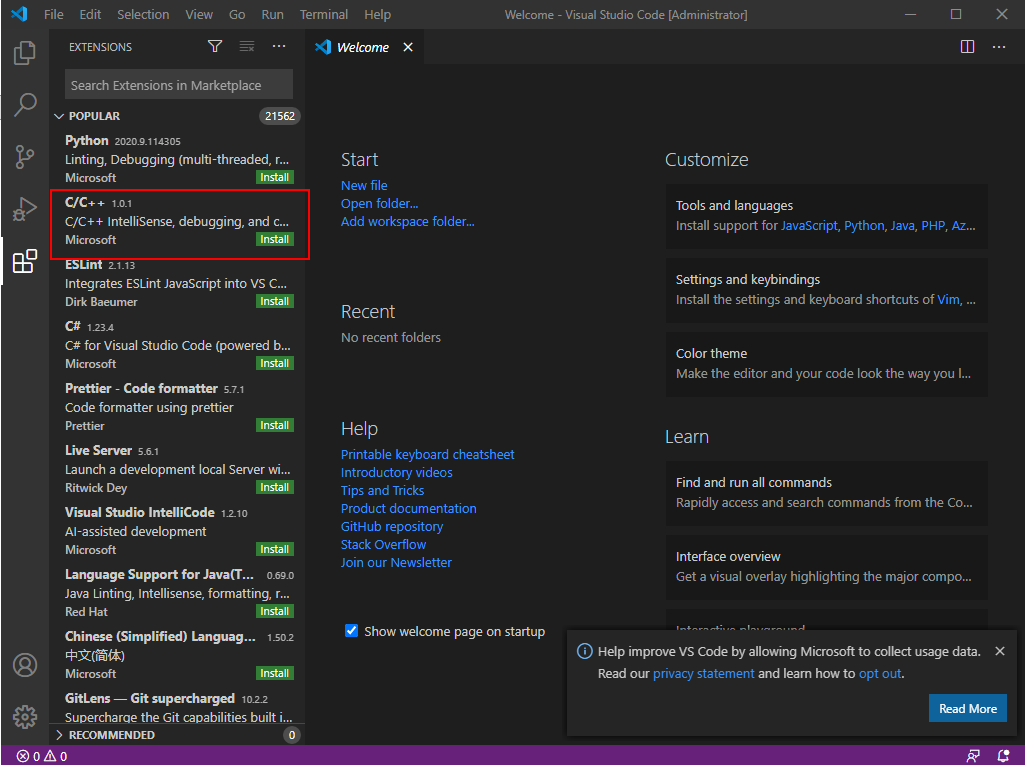
选中＂添加到PATH（重启后生效）＂复选框，勾选此选项后可不再配置环境变量直接使用。



单击＂完成＂，结束安装。

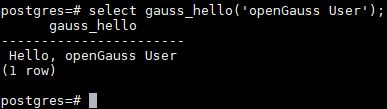


安装openGauss所需的开发语言C/C++。



# 二次开发（以实现gauss\_hello函数为例）

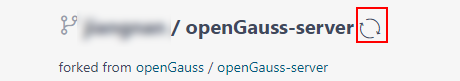
本章作为演示，为openGauss数据库新增一个系统函数gauss\_hello(name)，该函数能返回格式为＂Hello,<name>＂的提示信息，如下图：



## 将个人远程仓库内容更新至本地

（可选）将主仓库的内容更新至个人远程仓库。

如果非首次修改本地文件，建议您在开源社区网站的个人仓库页面，单击如下图标将主仓库的内容更新至个人远程仓库，使远程仓库与本地仓库保持一致。如下图所示：



在本地Git工具执行如下命令，将个人远程仓库内容更新至本地仓库。

git fetch upstream

git merge upstream/master



## 修改本地文件

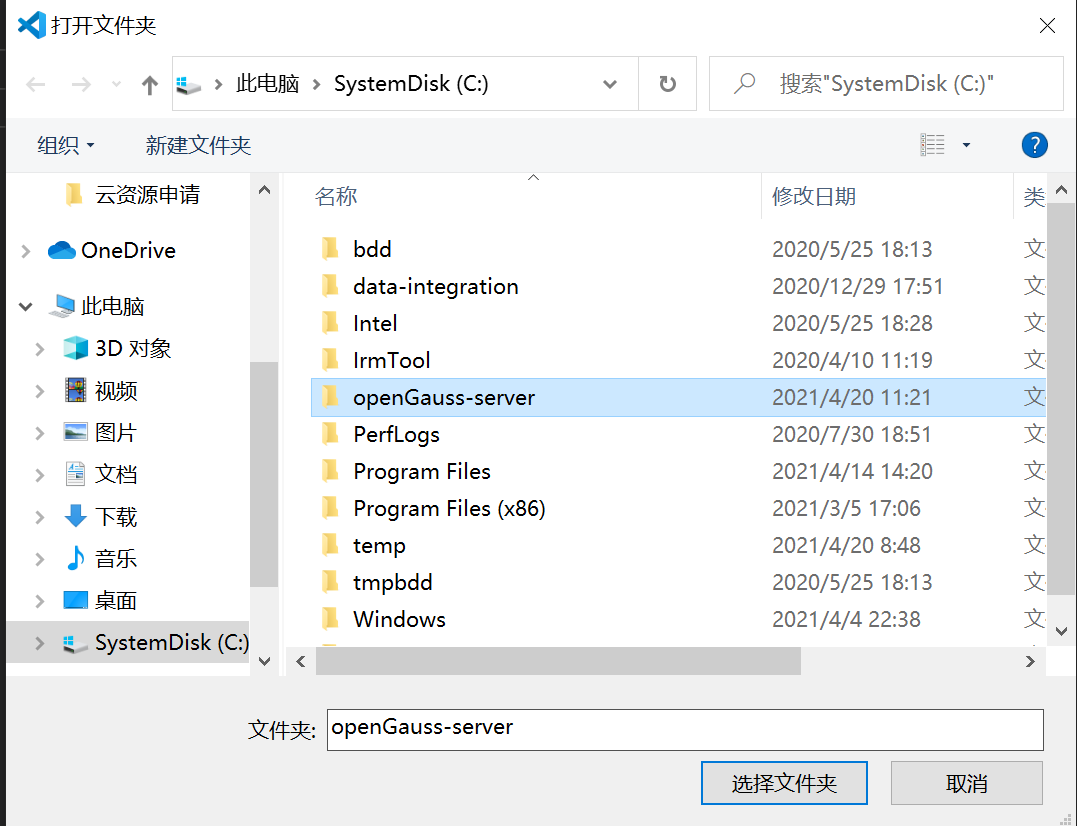
以实现gauss\_hello函数为例，需执行如下步骤。

使用VSCode导入数据库本地文件夹。

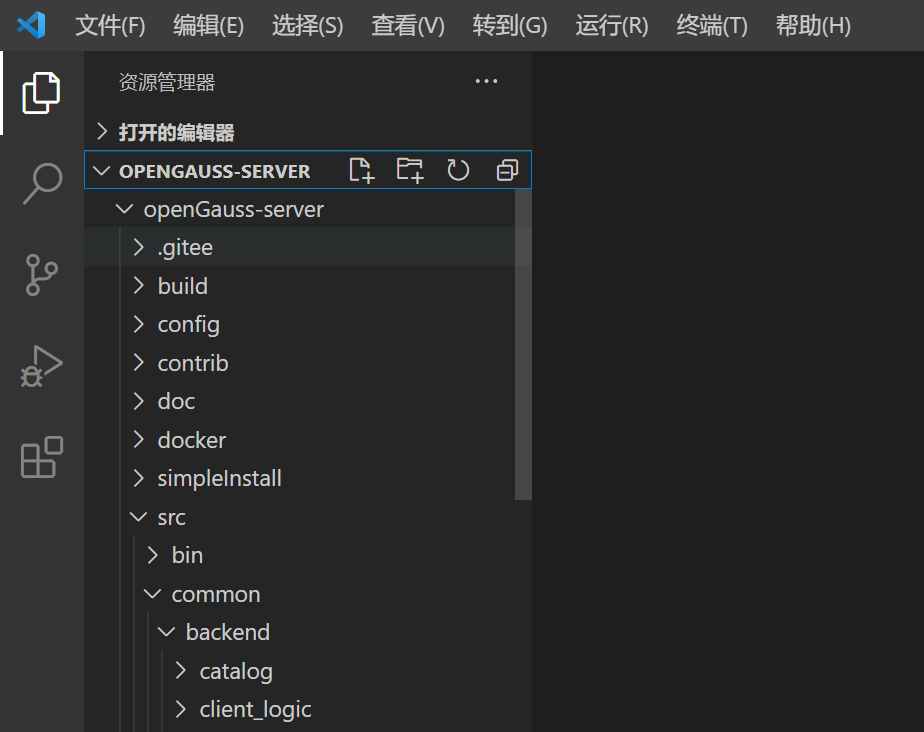
单击文件然后选择打开文件夹。



选择打开openGauss-server文件夹。



导入成功。



加入gauss\_hello函数实现。

修改本地文件src/common/backend/utils/adt/varlena.cpp，在文件末尾加入如下代码：

Datum gauss\_hello(PG\_FUNCTION\_ARGS)

{

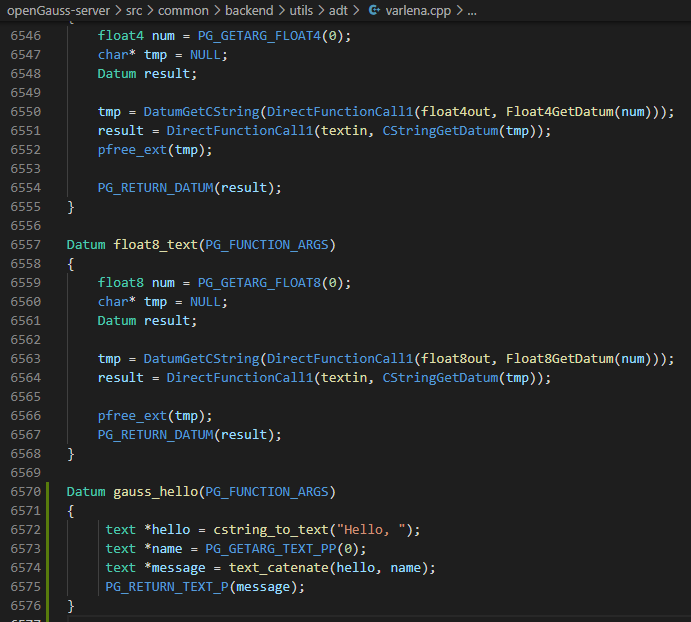
text \*hello = cstring\_to\_text("Hello, ");

text \*name = PG\_GETARG\_TEXT\_PP(0);

text \*message = text\_catenate(hello, name);

PG\_RETURN\_TEXT\_P(message);

}

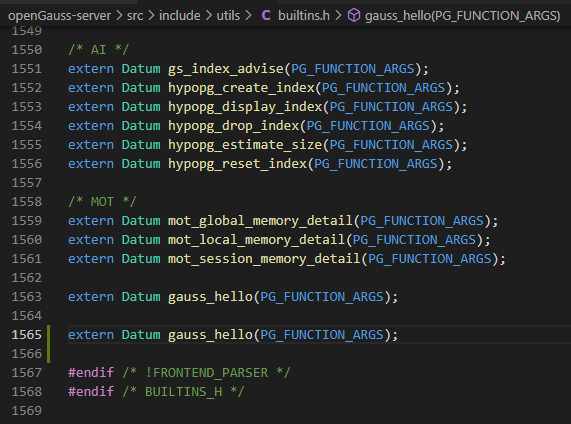


添加完成后点击保存，确认修改已经保存。

加入gauss\_hello函数声明。

修改本地文件src/include/utils/builtins.h，在文件末尾＂#endif＂前加入如下代码：

extern Datum gauss\_hello(PG\_FUNCTION\_ARGS);



添加完成后点击保存，确认修改已经保存。

将gauss\_hello函数注册到系统表。

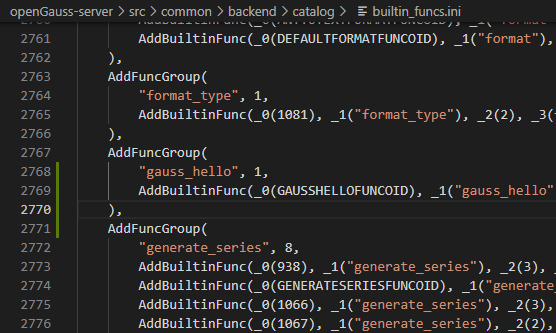
修改本地文件src/common/backend/catalog/builtin\_funcs.ini，在文件中按照函数名排序将如下代码加入合适位置：

AddFuncGroup(

"gauss\_hello", 1,

AddBuiltinFunc(\_0(GAUSSHELLOFUNCOID), \_1("gauss\_hello"), \_2(1), \_3(true), \_4(false), \_5(gauss\_hello), \_6(25), \_7(PG\_CATALOG\_NAMESPACE), \_8(BOOTSTRAP\_SUPERUSERID), \_9(INTERNALlanguageId), \_10(1), \_11(0), \_12(0), \_13(0), \_14(false), \_15(false), \_16(false), \_17(false), \_18('i'), \_19(0), \_20(1, 25), \_21(NULL), \_22(NULL), \_23(NULL), \_24(NULL), \_25("gauss\_hello"), \_26(NULL), \_27(NULL), \_28(NULL), \_29(0), \_30(false), \_31(NULL), \_32(false), \_33(NULL), \_34('f'))

),

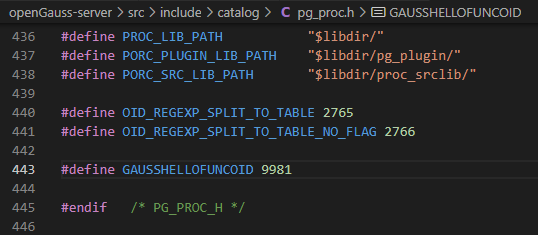


添加完成后点击保存，确认修改已经保存。

声明常量GAUSSHELLOFUNCOID。

修改本地文件src/include/catalog/pg\_proc.h，文件末尾＂#endif＂前加入如下代码，声明常量GAUSSHELLOFUNCOID，该常量将步骤3中的代码引用，作为gauss\_hello函数在系统表中注册时所使用的oid，如下代码以oid为9981为例：

#define GAUSSHELLOFUNCOID 9981



添加完成后点击保存，确认修改已经保存。

注意：

GAUSSHELLOFUNCOID常量须定义为一个尚未被其他函数占用的整数，否则数据库初始化时会因无法为oid建立唯一索引而初始化失败。

## 提交本地修改到个人远程仓库

在cmd中先进入修改文件所在的文件夹目录，然后在目录下打开Git工具。

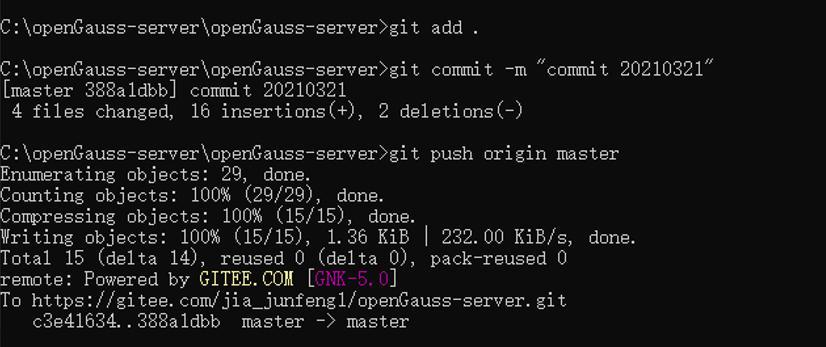
cd C:\openGauss-server\openGauss-server

git add .

接着在Git工具中执行如下命令，将本地修改提交至个人远程仓库。

git commit -m "20210321"  **----#提交原因**

git push origin master



# 编译

## 编译环境要求

### 操作系统环境要求

操作系统环境要求如下表。

|  |  |
| --- | --- |
| CPU | 操作系统要求 |
| x86架构 | CentOS 7.6  openEuler 20.3 LTS |
| ARM架构 | openEuler 20.3 LTS  麒麟V10 |

下文章节中以使用华为云ECS服务的鲲鹏服务器（ARM架构+ openEuler操作系统）为例进行介绍。

您也可以自行选择在本地PC机安装虚拟机进行操作（x86架构），详细步骤可参考《在虚拟机+CentOS上安装部署openGauss数据库指导手册》或《在虚拟机+openEuler上安装部署openGauss数据库指导手册》。

### 软件依赖要求

openGauss的软件依赖要求如下表所示：

|  |  |  |
| --- | --- | --- |
| 所需软件 | 建议版本 | 安装命令 |
| libaio-devel | 0.3.109-13 | yum install -y autoconf gcc gcc-c++ libaio-devel flex bison ncurses-devel glibc-devel patch readline readline-devel dkms python3-devel pam-devel libffi-devel automake byacc cmake openssl-devel libtool-devel |
| flex | 2.5.31 以上 |
| bison | 2.7-4 |
| ncurses-devel | 5.9-13.20130511 |
| Glibc-devel | 2.17-111 |
| patch | 2.7.1-10 |
| readline-devel | 7.0-13 |
| python3-devel | 3 |
| pam-devel | 1.1.8-1.3.1 |
| libffi-devel | 3.1 |
| automake | 1.13.4 |
| byacc | 1.9 |
| cmake | 3.19.2 |
| openssl-devel | 1.1.1 |
| libtool-devel | 2.4.2及以上 |
| lsb\_release | 4.1 |

## 编译环境准备（使用代金券购买华为云ECS服务）

### 登录华为云

进入华为云官网。

华为云官网：https://www.huaweicloud.com/，进入华为云官网，单击登录。



输入账号名和密码，单击登录。



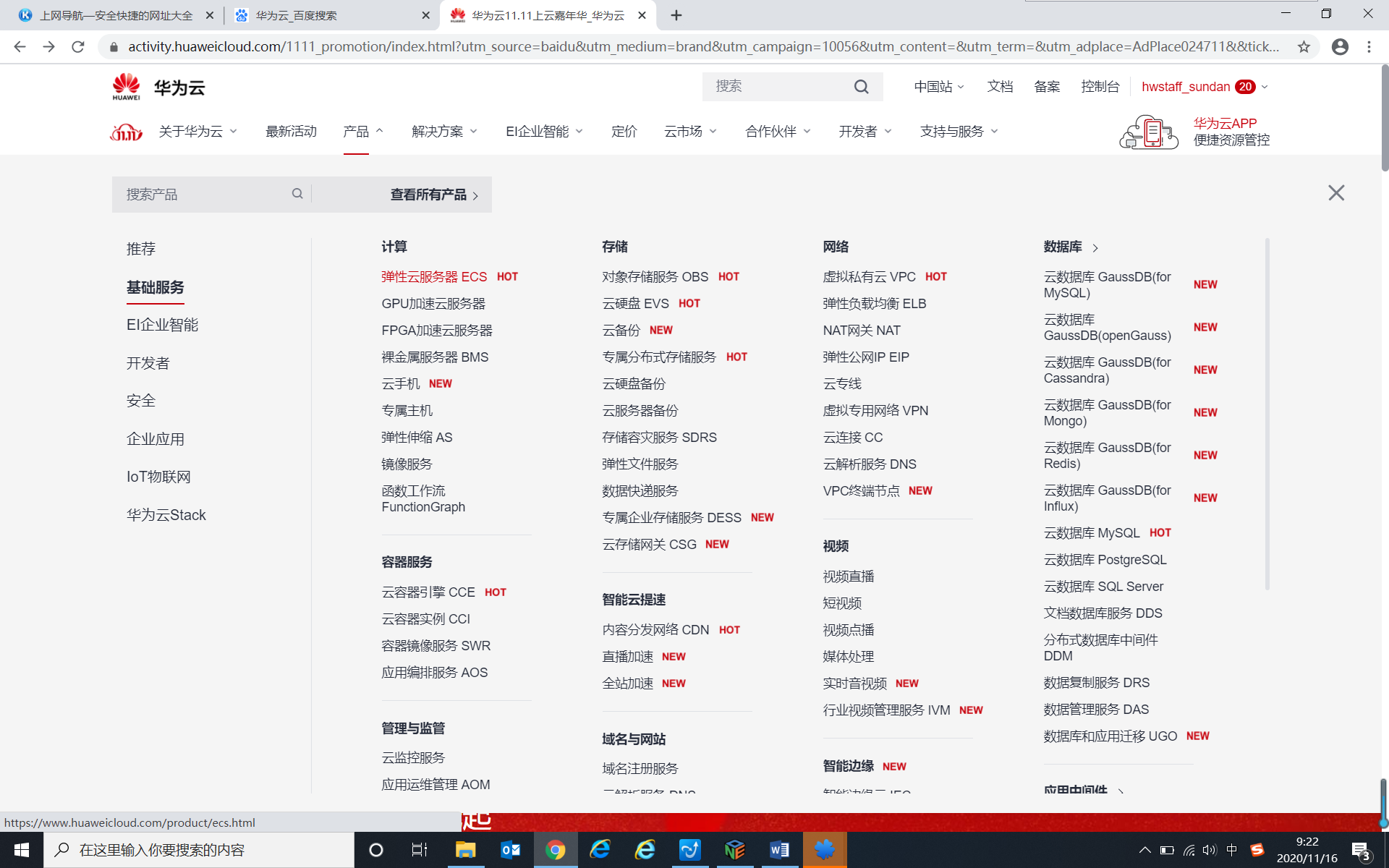
如果还没有注册，单击免费注册，按步骤进行注册与实名制认证后进行登录。

说明：

请进行实名制认证后再执行下述操作。

### 购买弹性云服务器ECS

在华为云主页（<https://www.huaweicloud.com/>）点击产品，选择基础服务，再选择＂弹性云服务器ECS＂。



**①**

**③**

**②**

进入弹性云服务器ECS购买界面。



单击＂立即购买＂，并自定义基础配置。。

|  |  |
| --- | --- |
| 配置选项 | 配置值 |
| 计费模式 | 按需计费（一定要选按需计费，注意配置费用） |
| 区域 | 华北-北京四（推荐，其他区域可能会导致无法选择openEuler公共镜像） |
| CPU架构 | 鲲鹏计算 |
| 规格 | 鲲鹏通用计算增强型 8vCPUs|32GiB  说明：该配置为推荐配置，若使用4vCPUs|8GB，编译用时会更长且编译报错比较多，不建议使用小于8vCPUs|32GiB的配置；  代金券金额充足可选用12vCPUs|48GiB的配置，编译时长更短。 |
| 镜像 | 公共镜像：  openEuler  openEuler 20.03 64bit with ARM(40GiB) |





其余默认即可，单击＂下一步网络配置＂。

自定义购买进行网路配置。

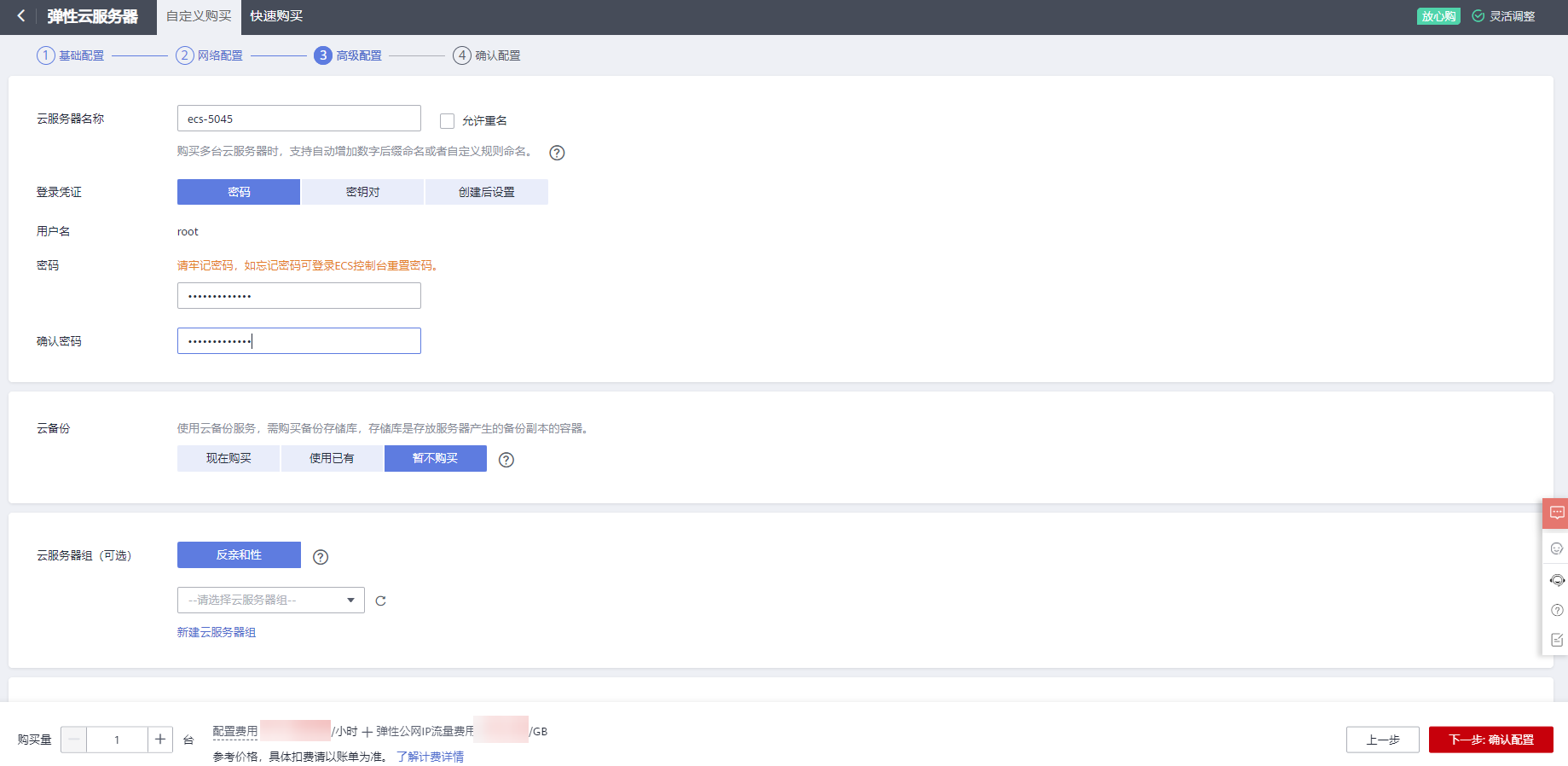
|  |  |
| --- | --- |
| 配置选项 | 配置值 |
| 网络 | vpc-default（192.168.0.0/16）（选现有默认网络即可） |
| 弹性公网IP | 现在购买 |
| 公网带宽 | 按流量计费 |
| 带宽大小 | 5 |





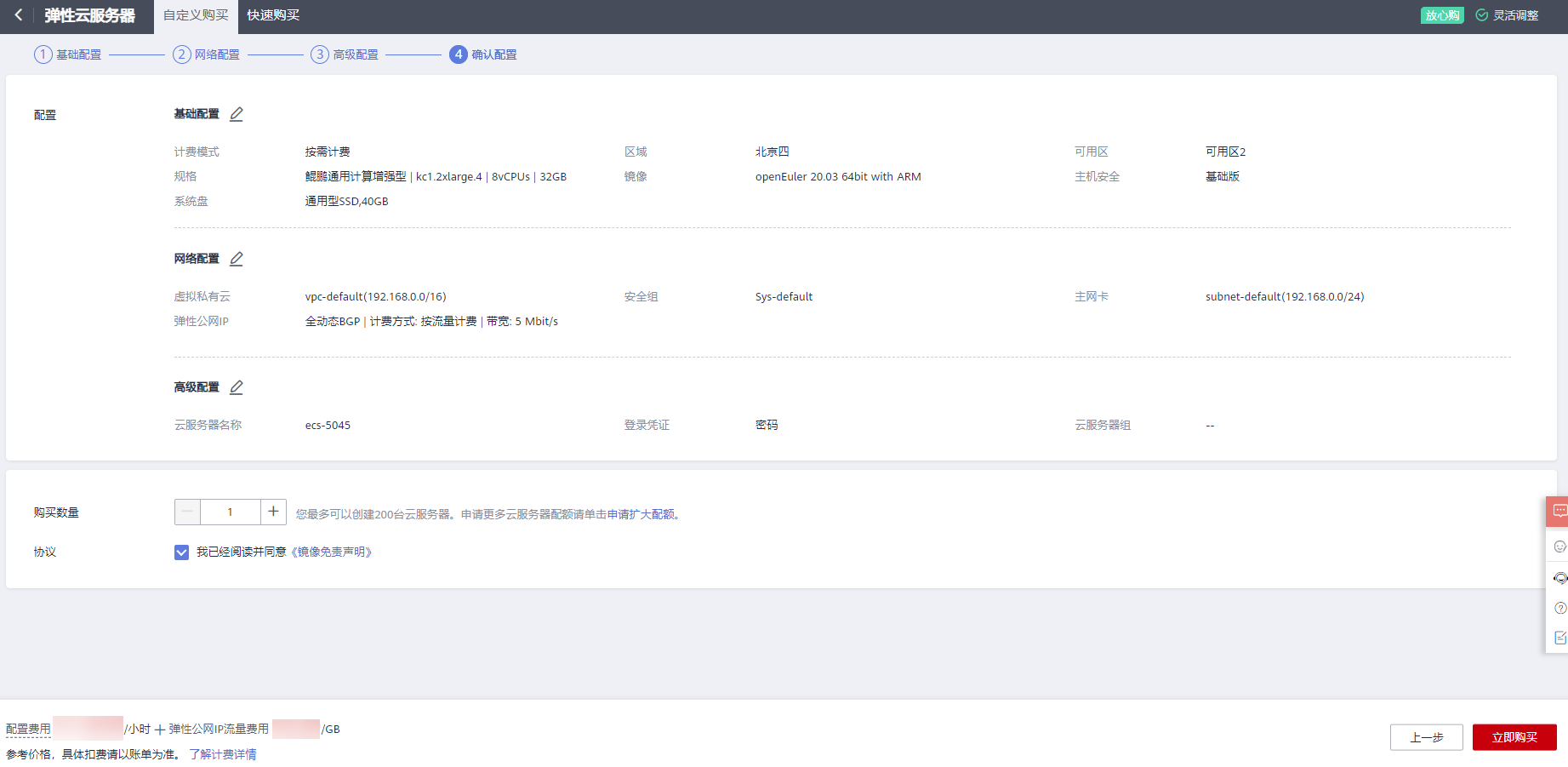
其余默认即可，单击＂下一步高级配置＂。

自定义购买进行高级配置。



记住用户名为root，然后输入自定义密码和确认密码，其余默认即可，单击＂下一步确认设置＂。

确认配置购买成功。



确认设置信息，尤其是配置费用，然后勾选协议＂我已经阅读并同意《镜像免责声明》＂，单击＂立即购买＂。



查看云服务器列表。



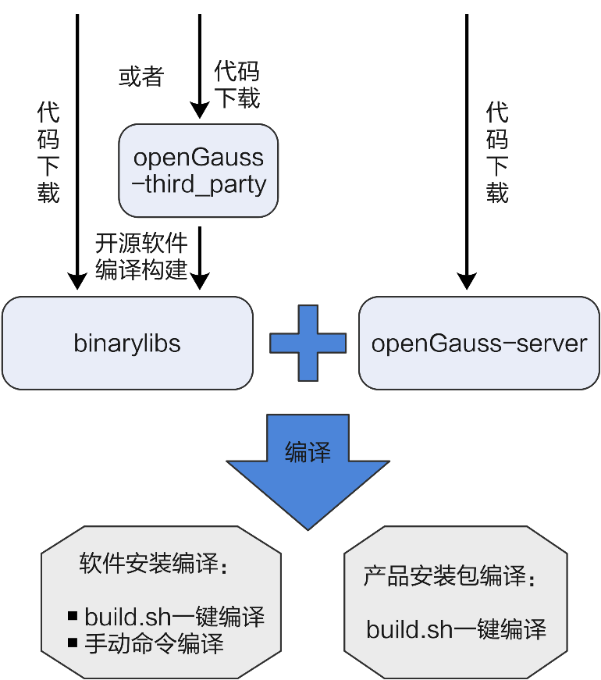
等待几分钟后，状态为＂运行中＂则表示购买成功！

注意：本次购买鲲鹏服务器价格为公测价格，具体价格以华为云官网为准。

## 版本编译

openGauss的编译和生成安装包的过程已有一个一键式的脚本build.sh，用户可以通过build.sh脚本更加方便地进行编译操作。

本节介绍openGauss使用一键式脚本编译的操作步骤，编译流程如下图所示：



### 代码下载

登录购买的ECS实例。

在弹性云服务列表上，单击服务器后面的＂远程登录＂操作，以root用户登录至服务器。



在Linux环境下安装软件依赖包。

[root@ecs-5045 ~]# yum install -y autoconf gcc gcc-c++ libaio-devel flex bison ncurses-devel glibc-devel patch readline readline-devel dkms python3-devel pam-devel libffi-devel automake byacc cmake openssl-devel libtool-devel

在Linux环境下安装Git工具并配置Git环境。

[root@ecs-5045 ~]# yum install git -y ---安装git工具

[root@ecs-5045 ~]# git config --global user.name "**Your Name**" --- 配置git用户名

[root@ecs-5045 ~]# git config --global user.email ["**email@example.com**"](mailto:%22email@example.com%22) --- 配置git全局邮件账号

Your Name为自己的git用户名，请替换为实际自己实际用户名。

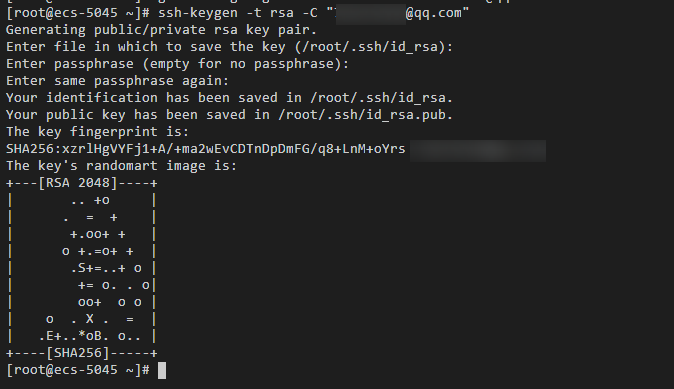
email@example.com为注册Gitee的主邮箱，请替换为实际自己实际主邮箱。

生成Git公钥和私钥。

[root@ecs-5045 ~]# ssh-keygen -t rsa -C ["**email@example.com**"](mailto:%22email@example.com%22) --- 生成git公钥和私钥

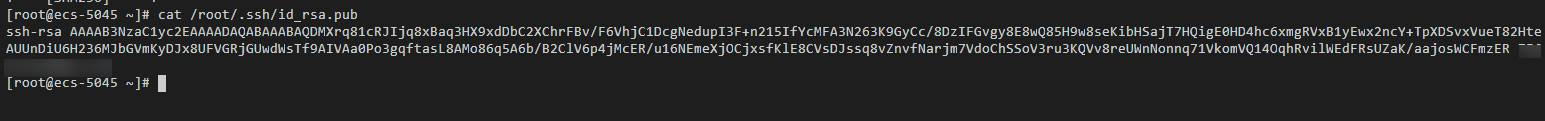
email@example.com为注册Gitee的主邮箱，请替换为自己实际主邮箱。

执行结果如下：

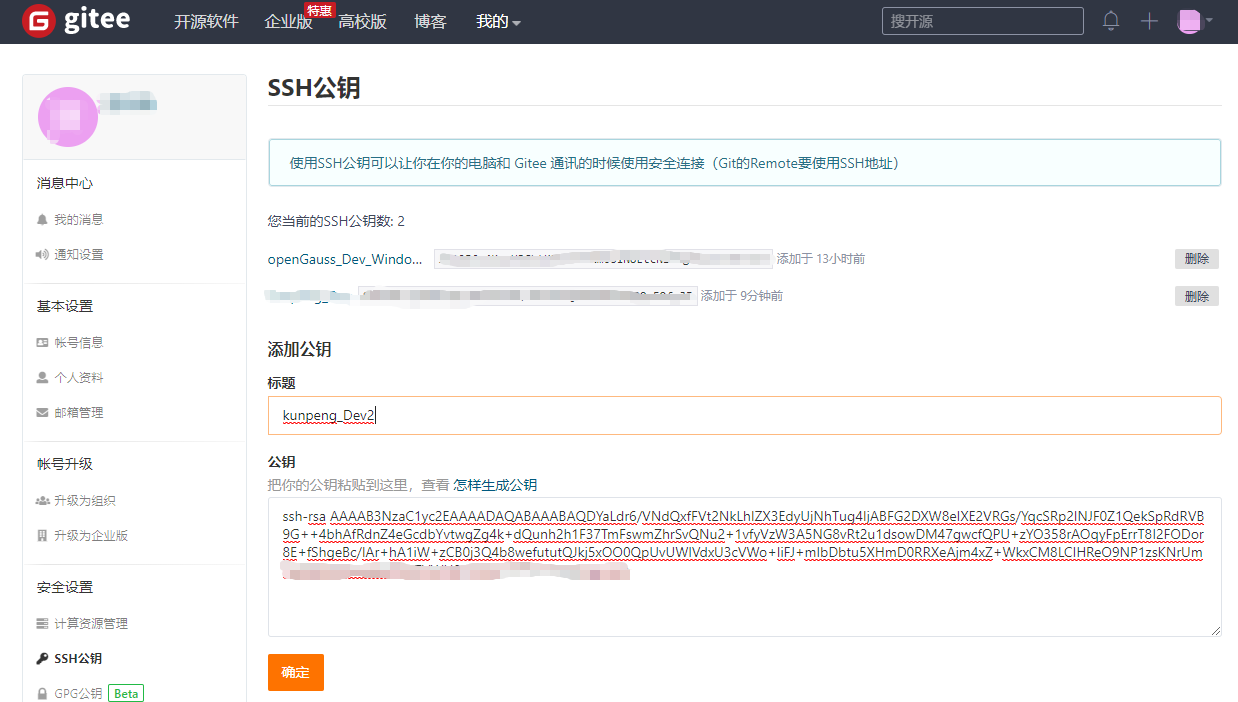


/root/.ssh/id\_rsa.pub为生成的Git公钥文件，读取该文件内容。

[root@ecs-5045 ~]# cat /root/.ssh/id\_rsa.pub



复制出上面文件内容，并在个人Gitee账户中添加此SSH公钥。



用Git工具下载openGauss-server和openGauss-OM代码。

先创建本地代码仓库存放目录/openGauss-lab/soft。

[root@ecs-5045 ~]# mkdir -p /openGauss-lab/soft

[root@ecs-5045 ~]# cd /openGauss-lab/soft

[root@ecs-5045 soft]#

openGauss-server是server代码仓库，openGauss-OM是工具代码仓库，两者均用于后续安装openGauss。

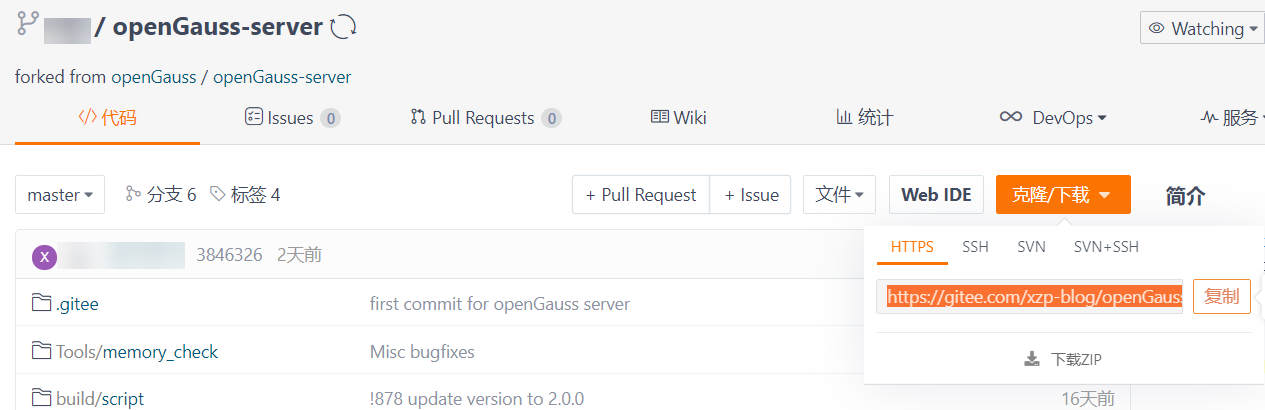
[root@ecs-5045 soft]# git clone **[git ssh address]** openGauss-server ##下载个人server代码仓库

[root@ecs-5045 soft]# git clone **[git ssh address]** openGauss-OM ##下载个人OM代码仓库

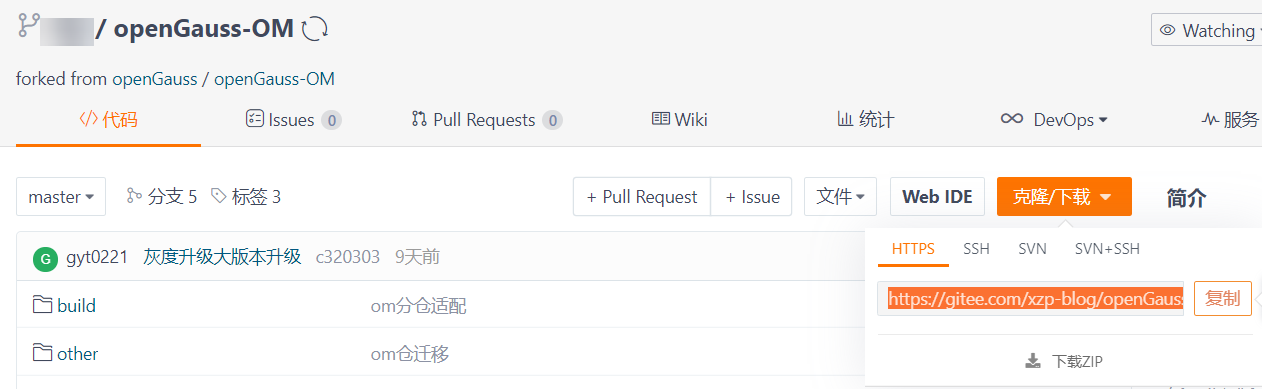
上述命令中：

*[git ssh address]*表示实际代码下载地址，即个人库地址，可在openGauss社区获取。

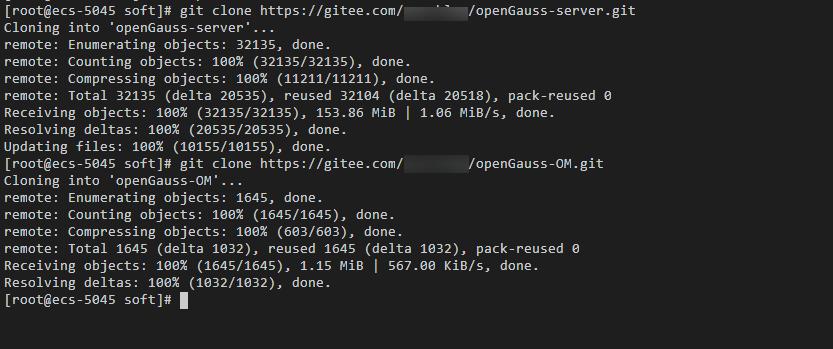
个人server代码仓库地址：



个人OM代码仓库地址：



执行下载结果如下：

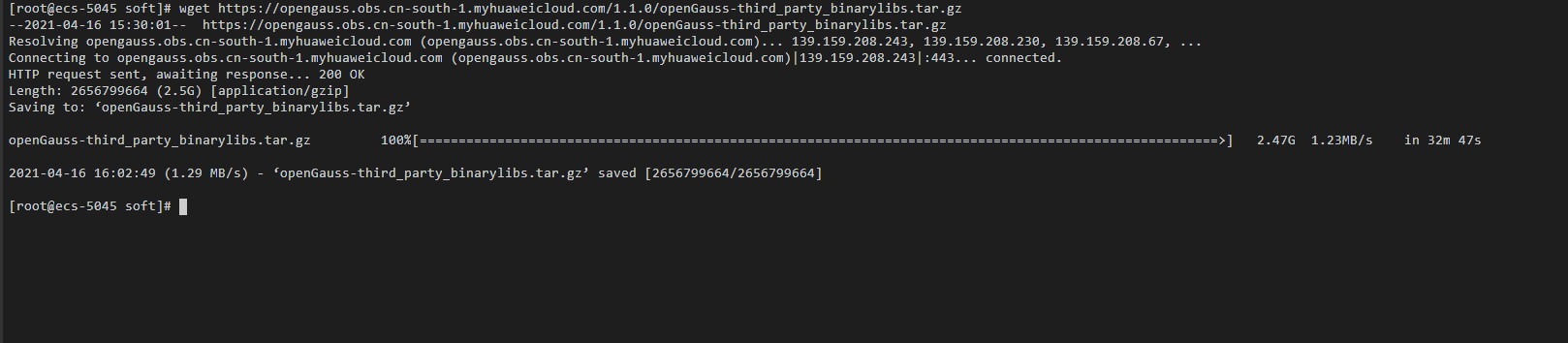


获取已编译好的第三方二进制代码。

该二进制文件用于openGauss-server代码的编译。

[root@ecs-5045 soft]# wget <https://opengauss.obs.cn-south-1.myhuaweicloud.com/1.1.0/openGauss-third_party_binarylibs.tar.gz>

下载项进度均显示为100%时表示下载成功，具体如下：



解压并移动该二进制文件。

[root@ecs-5045 soft]# tar -zxf openGauss-third\_party\_binarylibs.tar.gz

数分钟后解压成功。

[root@ecs-5045 soft]# mv openGauss-third\_party\_binarylibs binarylibs

### 软件安装编译

软件安装编译即将代码编译生成软件，并将软件安装到机器上。

openGauss提供一键式编译脚本＂openGauss-server/build.sh＂。＂build.sh＂是编译过程中的重要脚本工具，可快速进行代码编译和打包。

详细参数选项如下表所示。

说明：下表仅供了解，参数值无需自己选择，操作命令按步骤中执行即可。

| **功能选项** | **缺省值** | **参数** | **说明** |
| --- | --- | --- | --- |
| -h | 不使用此选项 | - | 帮助菜单。 |
| -m | release | [debug | release | memcheck] | 选择编译目标版本。有三个目标版本可以选择：  release：生成release版本的二进制程序。此版本编译时，通过配置GCC（GNU Compiler Collection，GNU编译器集）高级优化选项，去除内核调试代码。此选项通常在生产环境或性能测试环境中使用。  debug：表示生成debug版本的二进制程序。此版本编译时，增加了内核代码调试功能，一般用于开发自测环境。  memcheck：表示生成memcheck版本的二进制程序。此版本编译时，在debug版本的基础上增加了ASAN功能，用于定位内存问题。 |
| -3rd | ${代码路径}/binarylibs | [binarylibs path] | 指定binarylibs的路径，需绝对路径。默认设置为当前代码文件夹下存在binarylibs，因此如果binarylibs被移至openGauss-server中，或者在openGauss-server中创建了到binarylibs的软链接，则不需要指定此参数。 |
| -pkg | 不使用此功能 | - | 将代码编译结果压缩封装成安装包。 |
| -nopt | 不使用此功能 | - | 如果使用此功能，则对鲲鹏平台的相关CPU不进行优化。 |

该脚本中的每个选项都有一个默认值。

#### 前提条件

已按照搭建编译环境的要求准备好相关软硬件，并且已经下载了原码；

了解 build.sh脚本的参数选项和功能；

代码环境干净，没有以前编译生成的文件。

#### 编译openGauss-server

进入到软件代码编译脚本目录。

[root@ecs-5045 soft]# cd /openGauss-lab/soft/openGauss-server

编译安装openGauss-server。

[root@ecs-5045 openGauss-server] # sh build.sh -m release -3rd /openGauss-lab/soft/binarylibs -pkg

该步骤执行大概需要15-30分钟，与主机规格有关，请耐心等待。

显示如下内容，表示安装包编译成功。

…………………………………….

make server(all) package success!

packaging libpq...

success!

packaging tools...

success!

Begin to install upgrade\_sql files...

Successfully packaged upgrade\_sql files.

End package opengauss.

now, all packages has finished!

生成的安装包存放在＂/openGauss-lab/soft/openGauss-server/output＂目录下，编译结果：

openGauss-2.0.0-openEuler-64bit-Libpq.tar.gz

openGauss-2.0.0-openEuler-64bit.sha256

openGauss-2.0.0-openEuler-64bit-symbol.tar.gz

openGauss-2.0.0-openEuler-64bit.tar.bz2 -----此包大概有89M

openGauss-2.0.0-openEuler-64bit-toosl.tar.gz

upgrade\_sql.sha256

upgrade\_sql.tar.gz

[root@ecs-5045 openGauss-server]# cd /openGauss-lab/soft/openGauss-server/output

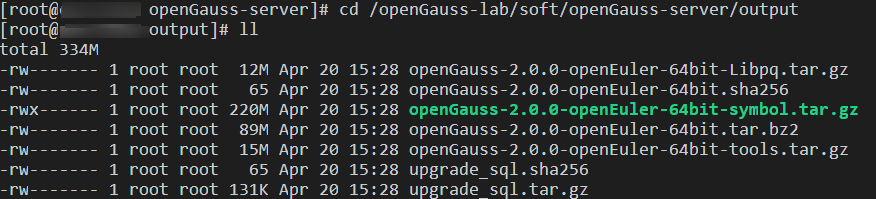
[root@ecs-5045 output]# ll

total 334M

-rw------- 1 root root 12M Apr 20 15:28 openGauss-2.0.0-openEuler-64bit-Libpq.tar.gz

-rw------- 1 root root 65 Apr 20 15:28 openGauss-2.0.0-openEuler-64bit.sha256

如下图：



编译日志为＂./build/script/makemppdb\_pkg.log＂，安装包打包日志为＂./build/script/make\_package.log＂。

#### 编译openGauss-OM

安装openGauss时需要OM工具，因此需要编译openGauss-OM。

进入到软件代码编译脚本目录。

[root@ecs-5045 soft]# cd /openGauss-lab/soft/openGauss-OM

编译安装OM。

[root@ecs-5045 openGauss-OM] # sh build.sh -3rd /openGauss-lab/soft/binarylibs

显示如下内容，表示安装包编译成功。

Everything is ready

success!

编译结果路径：/openGauss-lab/soft/openGauss-OM/package/。

编译结果：

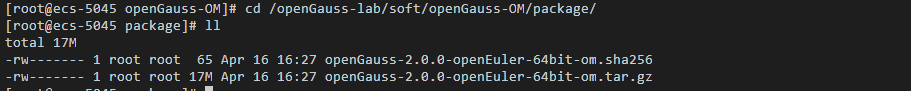
[root@ecs-5045 openGauss-OM]# cd /openGauss-lab/soft/openGauss-OM/package/

[root@ecs-5045 package]# ll

total 17M

-rw------- 1 root root 65 Apr 16 16:27 openGauss-2.0.0-openEuler-64bit-om.sha256

-rw------- 1 root root 17M Apr 16 16:27 openGauss-2.0.0-openEuler-64bit-om.tar.gz

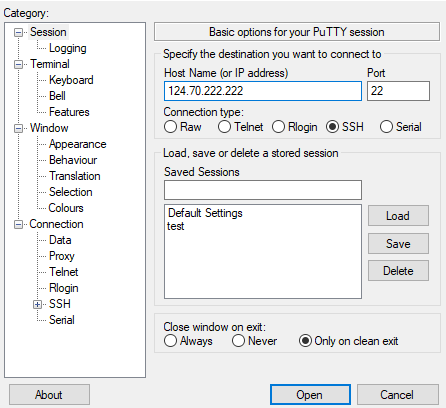


# 验证二次开发结果

## 安装编译后的openGauss安装包

### 修改操作系统配置

为了操作方便，也可以使用SSH工具（比如：PuTTY等）从本地电脑通过配置弹性云服务器的弹性公网IP地址（如：124.70.222.222）来连接ECS，并使用 root 用户来登录。



#### 设置字符集参数

将各数据库节点的字符集设置为相同的字符集，可以在/etc/profile文件中添加"export LANG=XXX"（XXX为Unicode编码）。

在/etc/profile文件中添加"export LANG= en\_US.UTF‐8"。

[root@ecs-5045 ~]# cat >>/etc/profile<<EOF

export LANG=en\_US.UTF‐8

EOF

输入如下命令，使修改生效。

[root@ecs-5045 ~]# source /etc/profile

#### 修改Python版本

之后安装过程中openGauss用户互信，openEuler服务器需要用到Python-3.7.x命令，但是默认Python版本为Python-2.7.x，所以需要切换Python版本。

进入/usr/bin文件，备份Python文件。

[root@ecs-5045 ~]# cd /usr/bin

备份python文件。

[root@ecs-5045 bin] # mv python python.bak

建立python3软连接。

[root@ecs-5045 bin] # ln -s python3 /usr/bin/python

验证Python版本。

[root@ecs-5045 bin] # python -V

显示如下，说明切换成功：

Python 3.7.4

### 规划数据库安装包目录

登录ECS，规划存放安装包的目录（/opt/software/openGauss）并创建，并将内容设置为当前用户可读可执行。

[root@ecs-5045 bin]# mkdir -p /opt/software/openGauss

[root@ecs-5045 bin]# chmod 755 -R /opt/software

说明：

不建议把安装包的存放目录规划到openGauss用户的home目录或其子目录下，可能导致权限问题。

openGauss用户须具有/opt/software/openGauss目录的读写权限。

将数据库安装包放置到规划目录下。

[root@ecs-5045 bin]# cp /openGauss-lab/soft/openGauss-server/output/\* /opt/software/openGauss

[root@ecs-5045 bin]# cp /openGauss-lab/soft/openGauss-OM/package/\* /opt/software/openGauss

说明：/openGauss-lab/soft/openGauss-server/output路径是编译openGauss-server后安装包的默认存放路径；

/openGauss-lab/soft/openGauss-OM/package是编译openGauss-OM后安装包的默认存放路径。

### 创建XML配置文件

安装openGauss前需要创建XML文件。XML文件包含部署openGauss的服务器信息、安装路径、IP地址以及端口号等。用于告知openGauss如何部署。用户需根据不同场合配置对应的XML文件。

以单节点配置的方案为例，说明如何创建XML配置文件。

以root用户登录待安装openGauss的主机，切换到存放安装包的目录。

[root@ecs-5045 bin]# cd /opt/software/openGauss

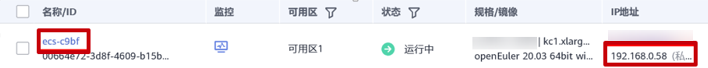
创建XML配置文件，用于数据库安装。

[root@ecs-5045 openGauss]# vi clusterconfig.xml

输入＂i＂进入INSERT模式，添加文本如下。

红色加粗字体内容为示例，可自行替换，其中＂ecs-5045＂是弹性云服务器的名称，＂192.168.0.58＂为弹性云服务器的IP地址（私有），其他value值可以不进行修改。

弹性云服务器名称及私有IP查看：



<?xml version="1.0" encoding="UTF-8"?>

<ROOT>

<CLUSTER>

<PARAM name="clusterName" value="dbCluster" />

<PARAM name="nodeNames" value="**ecs-5045**" />

<PARAM name="backIp1s" value="**192.168.0.58**"/>

<PARAM name="gaussdbAppPath" value="/opt/gaussdb/app" />

<PARAM name="gaussdbLogPath" value="/var/log/gaussdb" />

<PARAM name="gaussdbToolPath" value="/opt/huawei/wisequery" />

<PARAM name="corePath" value="/opt/opengauss/corefile"/>

<PARAM name="clusterType" value="single-inst"/>

</CLUSTER>

<DEVICELIST>

<DEVICE sn="1000001">

<PARAM name="name" value="**ecs-5045**"/>

<PARAM name="azName" value="AZ1"/>

<PARAM name="azPriority" value="1"/>

<PARAM name="backIp1" value="**192.168.0.58**"/>

<PARAM name="sshIp1" value="**192.168.0.58**"/>

<!--dbnode-->

<PARAM name="dataNum" value="1"/>

<PARAM name="dataPortBase" value="26000"/>

<PARAM name="dataNode1" value="/gaussdb/data/db1"/>

</DEVICE>

</DEVICELIST>

</ROOT>

按＂ESC＂键退出INSERT模式，接着输入＂:wq＂后回车退出编辑并保存文本。

配置文件参数附加说明

|  |  |
| --- | --- |
| 参数 | 说明 |
| clusterName | openGauss名称。 |
| nodeNames | openGauss中主机名称。 |
| backIp1s | 主机在后端存储网络中的IP地址（内网IP）。所有openGauss主机使用后端存储网络通讯。 |
| gaussdbAppPath | openGauss程序安装目录。此目录应满足如下要求：  • 磁盘空间>1GB  • 与数据库所需其它路径相互独立，没有包含关系。 |
| gaussdbLogPath | openGauss运行日志和操作日志存储目录。此目录应满足如下要求：  • 磁盘空间建议根据主机上的数据库节点数规划。数据库节点预留1GB空间的基础上，再适当预留冗余空间。  • 与openGauss所需其它路径相互独立，没有包含关系。  此路径可选。不指定的情况下，openGauss安装时会默认指定＂$GAUSSLOG/安装用户名＂作为日志目录。 |
| tmpdbPath | 数据库临时文件存放目录。  若不配置tmpdbPath，默认存放在/opt/huawei/wisequery/perfadm\_db目录下。 |
| gaussdbToolPath | openGauss系统工具目录，主要用于存放互信工具等。此目录应满足如下要求：  • 磁盘空间>100MB  • 固定目录，与数据库所需其它目录相互独立，没有包含关系。  此目录可选。不指定的情况下，openGauss安装时会默认指定＂/opt/huawei/wisequery＂作为数据库系统工具目录。 |
| corePath | openGauss core文件的指定目录。 |

须知：

＂opt/huawei/newsql/tools＂存放互信等工具，避免权限问题，不要把实例数据目录放在此目录下；

安装目录和数据目录需为空或者不存在，否则可能导致安装失败；

在对数据库节点的实例进行具体配置时，需确保配置的目录之间不相互耦合。即各个配置目录不关联，删除其中任意一个目录，不会级联删除其它目录。如gaussdbAppPath为"/opt/gaussdb/app"，gaussdbLogPath为"/opt/gaussdb/app/omm"。当gaussdbAppPath目录被删除时，会级联删除gaussdbLogPath目录，从而引起其它问题；

若需要安装脚本自动创建安装用户时各配置的目录需保证不与系统创建的默认用户目录耦合关联；

配置openGauss路径和实例路径时，路径中不能包含＂|＂,＂;＂,＂&＂,＂$＂,＂<＂,＂>＂,＂`＂,＂\\＂,＂'＂,＂\＂,＂{＂,＂}＂,＂(＂,＂)＂,＂[＂,＂]＂,＂~＂,＂\*＂,＂?＂特殊字符。

### 初始化安装环境

为了保证openGauss的正确安装，请首先对主机环境进行配置。

#### 准备安装用户及环境

创建完openGauss配置文件后，在执行安装前，为了后续能以最小权限进行安装及openGauss管理操作，保证系统安全性，需要运行安装前置脚本gs\_preinstall准备好安装用户及环境。

#### 前提条件

已完成安装准备的所有任务。

#### 注意事项

用户需要检查上层目录权限，保证安装用户对安装包和配置文件目录读写执行的权限；

xml文件中主机的名称与IP映射配置正确；

只能使用root用户执行gs\_preinstall命令。

#### 操作步骤

修改performance.sh文件。

使用vi打开文件＂/etc/profile.d/performance.sh＂。

[root@ecs-5045 openGauss]# vi /etc/profile.d/performance.sh

输入＂i＂，进入INSERT模式。用#注释sysctl -w vm.min\_free\_kbytes=112640 &> /dev/null这行。

CPUNO=`cat /proc/cpuinfo|grep processor|wc -l`

export GOMP\_CPU\_AFFINITY=0-$[CPUNO - 1]

#sysctl -w vm.min\_free\_kbytes=112640 &> /dev/null

sysctl -w vm.dirty\_ratio=60 &> /dev/null

sysctl -w kernel.sched\_autogroup\_enabled=0 &> /dev/null

按＂ESC＂键退出INSERT模式，接着输入＂:wq＂后回车退出编辑并保存文本。

为确保OpenSSL版本正确，执行预安装前需要加载安装包中lib库。

执行命令如下。

[root@ecs-5045 openGauss]# vi /etc/profile

输入i，进入INSERT模式，在文件的底部添加如下代码，加载安装包中lib库，其中packagePath为用户安装包放置的路径，本示例中为/opt/software/openGauss。按下＂Esc＂退出INSERT模式，输入＂:wq＂后回车，保存后退出。

export packagePath=/opt/software/openGauss

export LD\_LIBRARY\_PATH=$packagePath/script/gspylib/clib:$LD\_LIBRARY\_PATH

配置完成后，输入如下命令，使设置生效。

[root@ecs-5045 openGauss]# source /etc/profile

在openGauss安装包所在的目录下，解压安装包。

执行以下命令进入安装包目录。

[root@ecs-5045 openGauss]# cd /opt/software/openGauss

解压openGauss-2.0.0-openEuler-64bit-om.tar.gz包。

[root@ecs-5045 openGauss]# tar -zxvf openGauss-2.0.0-openEuler-64bit-om.tar.gz

解压后，用ls命令查看如下。

[root@ecs-5045 openGauss]# ls

clusterconfig.xml

openGauss-2.0.0-openEuler-64bit-om.sha256

openGauss-2.0.0-openEuler-64bit-symbol.tar.gz script pgrade\_sql.tar.gz

openGauss-2.0.0-openEuler-64bit-om.tar.gz

openGauss-2.0.0-openEuler-64bit.tar.bz2 simpleInstall version.cfg

openGauss-2.0.0-openEuler-64bit-Libpq.tar.gz openGauss-2.0.0-openEuler-64bit.sha256 openGauss-2.0.0-openEuler-64bit-tools.tar.gz upgrade\_sql.sha256

安装包解压后，会在/opt/software/openGauss路径下自动生成script子目录，并且在script目录下生成gs\_preinstall等各种OM工具脚本。

使用gs\_preinstall准备好安装环境，切换到gs\_preinstall命令所在目录。

[root@ecs-5045 openGauss]# cd /opt/software/openGauss/script/

script中内容显示如下：

[root@ecs-5045 script]# ls

gs\_backup gs\_checkperf gs\_om gspylib gs\_uninstall \_\_init\_\_.py

gs\_check gs\_collector gs\_postuninstall gs\_ssh gs\_upgradectl killall

gs\_checkos gs\_install gs\_preinstall gs\_sshexkey impl local

采用交互模式执行，并在执行过程中会创建root用户互信和openGauss用户互信。

[root@ecs-5045 script]# python gs\_preinstall -U omm -G dbgrp -X /opt/software/openGauss/clusterconfig.xml

命令中的omm为操作系统用户（同时omm也是openGauss的数据库管理员账号，在下面的5.1.5 章节中会创建），dbgrp为运行openGauss的操作系统用户的群组名称，/opt/software/openGauss/clusterconfig.xml为openGauss配置文件路径。在执行过程中，用户根据提示选择是否创建互信，具体如下：

此时会创建操作系统omm用户，并对omm创建trust，并设置密码，设置为Admin@123（建议用户自定义设置密码）。

Are you sure you want to create the user[omm] and create trust for it (yes/no)? yes

Please enter password for cluster user.

Password: --说明：此处输入密码时，屏幕上不会有任何反馈，不用担心，这是LINUX操作系统对密码的保护.

Please enter password for cluster user again.

Password: --说明：此处输入密码时，屏幕上不会有任何反馈，不用担心，这是LINUX操作系统对密码的保护.

Successfully created [omm] user on all nodes.

成功后显示为：

……

Setting finish flag.

Successfully set finish flag.

Preinstallation succeeded.

### 执行安装

按上述步骤准备好openGauss安装环境之后，按照启动安装过程部署openGauss。

#### 前提条件

已成功执行前置脚本gs\_preinstall；

服务器操作系统和网络均正常运行。

#### 操作步骤

修改文件权限。

[root@ecs-5045 script]# chmod -R 755 /opt/software/openGauss/script

登录到openGauss的主机，并切换到omm用户。

[root@ecs-5045 script]# su - omm

说明：

omm指的是前置脚本gs\_preinstall中-U参数指定的用户。

安装脚本gs\_install必须以前置脚本中指定的omm执行，否则，脚本执行会报错。

使用gs\_install安装openGauss。

执行以下命令进行安装：

gs\_install -X /opt/software/openGauss/clusterconfig.xml --gsinit-parameter="--encoding=UTF8" --dn-guc="max\_process\_memory=3GB" --dn-guc="shared\_buffers=256MB" --dn-guc="bulk\_write\_ring\_size=256MB" --dn-guc="cstore\_buffers=16MB"

具体如下：

[omm@ecs-c9bf ~]$ gs\_install -X /opt/software/openGauss/clusterconfig.xml --gsinit-parameter="--encoding=UTF8" --dn-guc="max\_process\_memory=3GB" --dn-guc="shared\_buffers=256MB" --dn-guc="bulk\_write\_ring\_size=256MB" --dn-guc="cstore\_buffers=16MB"

/opt/software/ openGauss/clusterconfig.xml为openGauss配置文件的路径。在执行过程中，用户需根据提示输入数据库管理员omm用户的密码，密码具有一定的复杂度，为保证用户正常使用该数据库，请记住输入的密码。

按照设置密码要求，比如设置密码为GaussDB@123（**建议用户自定义设置密码**）：

encrypt cipher and rand files for database.

Please enter password for database: --说明：此处输入密码时，屏幕上不会有任何反馈

Please repeat for database: --说明：此处输入密码时，屏幕上不会有任何反馈

begin to create CA cert files

设置的密码要符合复杂度要求：

最少包含8个字符；

不能和用户名和当前密码（ALTER）相同，或和当前密码反序；

至少包含大写字母（A-Z），小写字母（a-z），数字，非字母数字字符（限定为~!@#$%^&\*()-\_=+\|[{}];:,<.>/?）四类字符中的三类字符。

如果安装成功，显示如下：

……

Successfully deleted instances from all nodes.

Checking node configuration on all nodes.

Initializing instances on all nodes.

Updating instance configuration on all nodes.

Check consistence of memCheck and coresCheck on database nodes.

Configuring pg\_hba on all nodes.

Configuration is completed.

Successfully started cluster.

Successfully installed application.

end deploy..

### 安装生成的目录

安装后的目录及各目录下的文件说明请参见**下表**。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 项目目录说明 | 目录 | 子目录 | 说明 |
| 1 | 集群openGauss安装目录 | /opt/gaussdb/app | etc | cgroup工具配置文件。 |
| include | 存放数据库运行所需要的头文件。 |
| lib | 存放数据库的库文件的目录。 |
| share | 存放数据库运行所需要的公共文件，如配置文件模板。 |
| 2 | 集群openGauss数据目录 | /gaussdb/data | data\_dnxxx | DBnode实例的数据目录，其中主实例的目录名为＂data\_dnxxx＂，  备实例的为data\_dnSxxx。xxx代表DBnode编号。 |
| 3 | 集群openGauss日志目录 | /var/log/gaussdb/用户名 | bin | 二进制程序的日志目录。 |
| gs\_profile | 数据库内核性能日志目录。 |
| om | OM的日志目录。例如：  部分local脚本产生的日志，增删数据库节点接口的日志，gs\_om接口的日志，前置接口的日志，节点替换接口的日志等。 |
| pg\_audit | 数据库审计日志目录。 |
| pg\_log | 数据库节点实例的运行日志目录。 |
| 4 | 集群openGauss系统工具目录 | /opt/huawei/wisequery | script | 用于openGauss用户进行openGauss管理的脚本文件。 |
| lib | bin目录下的二进制文件依赖的库文件。 |

## 登录数据库验证结果

（可选）在数据库主节点服务器上，如果当前不在omm用户环境下，请切换至omm操作系统用户环境。

[root@ecs-5045 script]# su - omm

启动数据库服务（可选操作，安装完毕后默认启动，如未启动，请按此步骤启动）。

启动服务命令：

[omm@ecs-504 ~]$ gs\_om -t start

Starting cluster.

=========================================

Successfully started.

查看服务是否启动：

[omm@ecs-504 ~]$ gs\_om -t status

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

cluster\_name : dbCluster

cluster\_state : Normal

redistributing : No

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

连接数据库。

[omm@ecs-504 ~]$ gsql -d postgres -p 26000 -r

当结果显示为如下信息，则表示连接成功。

gsql ((openGauss 2.0.0 build 38463266) compiled at 2021-04-16 16:11:37 commit 0 last mr )

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

Type "help" for help.   
postgres=#

其中，postgres为openGauss安装完成后默认生成的数据库。初始可以连接到此数据库进行新数据库的创建。26000为数据库主节点的端口号，需根据openGauss的实际情况做替换。

说明：

gsql是openGauss数据库提供的命令行方式的数据库连接工具。

验证第3 章中开发的gauss\_hello函数是否新增成功。

postgres=# select gauss\_hello('openGauss User');

成功显示如下：

gauss\_hello

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

Hello, openGauss User

本实验结束。

# 附录一：Linux操作系统相关命令

Linux中的命令格式为：command [options] [arguments] 中括号表示可选的，即有些命令不需要选项也不需要参数，但有的命令在运行时需要多个选项或参数。

* options(选项)：选项是调整命令执行行为的开关，选项的不同决定了命令的显示结果不同。
* agruments(参数)：参数是指命令的作用对象。

## vi/vim

文本编辑器，若文件存在则是编辑，若不存在则是创建并编辑文本。

命令语法：

vim [参数]

参数说明：可编辑的文件名。

命令示例：

编辑名为clusterconfig的xml文本：

vim clusterconfig.xml

注：

vim编辑器有以下三种模式：

正常模式：其它模式下按Esc或Ctrl+[进入，左下角显示文件名或为空。

插入模式：正常模式下按i键进入，左下角显示--INSERT--。

可视模式：正常模式下按v键进入，左下角显示--VISUAL--。

退出命令（正常模式下）：

:wq 保存并退出。

:q! 强制退出并忽略所有更改。

:e! 放弃所有修改，并打开原有文件。

## cd

显示当前目录的名称，或切换当前的目录（打开指定目录）。

命令语法：

cd [参数]

参数说明：

无参数：切换用户当前目录。

. ：表示当前目录。

.. ：表示上一级目录。

~ ：表示home目录。

/ ：表示根目录。

命令示例：

切换到usr目录下的bin目录中。

cd /usr/bin

切换到用户home目录。

cd

切换到当前目录(cd后面接一个.)。

cd .

切换到当前目录上一级目录(cd后面接两个.)。

cd ..

切换到用户home目录。

cd ~

切换到根目录下。

cd /

注：切换目录需要理解绝对路径和相对路径这两个概念。

绝对路径：在Linux中，绝对路径是从/（即根目录）开始的，例如 /opt/software、/etc/profile, 如果目录以 / 就是绝对目录。

相对路径：是以 . 或 .. 开始的目录。 . 表示用户当前操作所在的位置，而 .. 表示上级目录。例如 ./gs\_om 表示当前目录下的文件或者目录。

## mv

文件或目录改名(move (rename) files)或将文件或目录移入其它位置，经常用来备份文件或者目录。

命令语法：

mv [选项] 参数1 参数2

常用选项：

-b：若需覆盖文件，则覆盖前先行备份。

参数说明：

参数1：源文件或目录。

参数2：目标文件或目录。

命令示例：

将文件python重命名为python.bak：

mv python python.bak

将/physical/backup目录下的所有文件和目录移到/data/dbn1目录下：

mv /physical/backup/\* /data/dbn1

## curl

在Linux中curl是一个利用URL规则在命令行下工作的文件传输工具。支持文件的上传和下载，是综合传输工具。

命令语法：

curl [选项] [URL]

常用选项：

-A/--user-agent <string>：设置用户代理发送给服务器；

-C/--continue-at <offset>：断点续转；

-D/--dump-header <file>：把header信息写入到该文件中；

-e/--referer：来源网址；

-o/--output：把输出写到该文件中；

-O/--remote-name：把输出写到该文件中，保留远程文件的文件名；

-s/--silent：静默模式。不输出任何东西；

-T/--upload-file <file>：上传文件；

-u/--user <user[:password]>：设置服务器的用户和密码；

-x/--proxy <host[:port]>：在给定的端口上使用HTTP代理；

-#/--progress-bar：进度条显示当前的传送状态。

参数说明：

URL：指定的文件传输URL地址。

命令示例：

将url(<https://mirrors.huaweicloud.com/repository/conf/CentOS-7-anon.repo)的内容保存到/etc/yum.repos.d/CentOS-Base.repo>文件中。

curl -o /etc/yum.repos.d/CentOS-Base.repo <https://mirrors.huaweicloud.com/repository/conf/CentOS-7-anon.repo>

如果在传输过程中掉线，可以使用-C的方式进行续传。

curl -C -O <https://mirrors.huaweicloud.com/repository/conf/CentOS-7-anon.repo>

## yum

Shell 前端软件包管理器。基于 RPM 包管理，能够从指定的服务器自动下载 RPM 包并且安装，可以自动处理依赖性关系，并且一次安装所有依赖的软体包，无须繁琐地一次次下载和安装。

命令语法：

yum [options] [command] [package ...]

常用选项：

-h：查看帮助；

-y：当安装过程提示选择全部为 "yes"；

-q：不显示安装的过程。

参数说明：

command：要进行的操作。

package：安装的包名。

命令示例：

列出所有可更新的软件清单命令。

yum check-update

更新所有软件命令。

yum update

列出所有可安装的软件清单命令。

yum list

安装指定的软件。

yum install -y libaio-devel flex bison ncurses-devel glibc.devel patch lsb\_release wget python3

## wget

wget是Linux下下载文件的最常用命令。wget支持HTTP，HTTPS和FTP协议，支持自动下载,即可以在用户退出系统后在后台执行,直到下载结束。

命令语法：

wget [选项] [URL]

常用选项：

-c：接着下载没下载完的文件；

-b：启动后转入后台执行；

-P：指定下载目录；

-O：变更下载文件名；

--ftp-user --ftp-password：使用FTP用户认证下载。

参数说明：

指定的文件下载URL地址。

命令示例：

下载openGauss数据库安装文件到当前文件夹：

wget [https://opengauss.obs.cn-south-1.myhuaweicloud.com/1.1.0/x86/openGauss-1.1.0-CentOS-64bit.tar.gz](https://opengauss.obs.cn-south-1.myhuaweicloud.com/1.0.0/x86/openGauss-1.0.0-CentOS-64bit.tar.gz)

使用wget断点续传：

wget –c [https://opengauss.obs.cn-south-1.myhuaweicloud.com/1.1.0/x86/openGauss-1.1.0-CentOS-64bit.tar.gz](https://opengauss.obs.cn-south-1.myhuaweicloud.com/1.0.0/x86/openGauss-1.0.0-CentOS-64bit.tar.gz)

## ln

为某一个文件在另外一个位置建立一个同步的链接（软硬链接，不带选项为硬链接）。

当需要在不同的目录，用到相同的文件时，就不需要在每一个需要的目录下都放一个必须相同的文件，我们只要在某个固定的目录，放上该文件，然后在其它的目录下用ln命令链接（link）它就可以，不必重复的占用磁盘空间。

命令语法：

ln [选项] 参数1 参数2

常用选项：

-b --删除，覆盖以前建立的链接；

-d --允许超级用户制作目录的硬链接；

-s --软链接(符号链接)。

参数说明：

参数1：源文件或目录。

参数2：被链接的文件或目录。

命令示例：

为python3文件创建软链接/usr/bin/python，如果python3丢失，/usr/bin/python将失效：

ln -s python3 /usr/bin/python

为python3创建硬链接/usr/bin/python，python3与/usr/bin/python的各项属性相同：

ln python3 /usr/bin/python

## mkdir

创建指定的名称的目录，要求创建目录的用户在当前目录中具有写权限，并且指定的目录名不能是当前目录中已有的目录。

命令语法：

mkdir [选项] [参数]

常用选项：

-p --可以是一个路径名称。此时若路径中的某些目录尚不存在,加上此选项后,系统将自动建立好那些尚不存在的目录,即一次可以建立多个目录（递归）；

-v --每次创建新目录都显示信息；

-m --设定权限<模式> (类似chmod)，而不是rwxrwxrwx减umask。

参数说明：

需要创建的目录。

命令示例：

创建一个空目录：

mkdir test

递归创建多个目录：

mkdir -p /opt/software/openGauss

创建权限为777的目录（目录的权限为rwxrwxrwx）：

mkdir –m 777 test

## chmod

更改文件权限。

命令语法：

chmod [选项] <mode> <file...>

常用选项：

-R, --以递归的方式对目前目录下的所有文件与子目录进行相同的权限变更。

参数说明：

mode：权限设定字串，详细格式如下 ：

[ugoa...][[+-=][rwxX]...][,...]，

其中，[ugoa...]：u 表示该档案的拥有者，g 表示与该档案的拥有者属于同一个群体(group)者，o 表示其他以外的人，a 表示所有（包含上面三者）；[+-=]：+ 表示增加权限，- 表示取消权限，= 表示唯一设定权限；[rwxX]：r 表示可读取，w 表示可写入，x 表示可执行，X表示只有当该档案是个子目录或者该档案已经被设定过为可执行。

file：文件列表（单个或者多个文件、文件夹）。

命令示例：

设置所有用户可读取文件 cluterconfig.xml：

chmod ugo+r cluterconfig.xml

或

chmod a+r cluterconfig.xml

设置当前目录下的所有档案与子目录皆设为任何人可读写：

chmod -R a+rw \*

数字权限使用格式：

这种使用方式中，规定数字4、2和1表示读、写、执行权限，即r=4,w=2,x=1。

例：rwx = 7（4+2+1）；rw = 6（4+2）；r-x = 5 （4+0+1）；r-- = 4（4+0+0）；--x = 1（0+0+1）；

每个文件都可以针对三个粒度，设置不同的rwx(读写执行)权限。即我们可以用三个8进制数字分别表示 拥有者 、群组 、其它组( u、 g 、o)的权限详情，并用chmod直接加三个8进制数字的方式直接改变文件权限。语法格式为 ：

chmod <abc> file...

其中，a，b，c各为一个数字，分别代表User、Group、及Other的权限，相当于简化版的chmod u=权限,g=权限,o=权限 file...，而此处的权限将用8进制的数字来表示User、Group、及Other的读、写、执行权限。

命令示例：

赋予cluterconfig.xml文件可读可写可执行权限（所有权限）：

chmod 777 cluterconfig.xml

赋予/opt/software/openGauss目录下所有文件及其子目录 用户所有权限组可读可执行权限，其他用户可读可执行权限：

chmod R 755 /opt/software/openGauss

## chown

利用 chown 将指定文件的拥有者改为指定的用户或组，用户可以是用户名或者用户ID；组可以是组名或者组ID；文件是以空格分开的要改变权限的文件列表，支持通配符。只有系统管理者(root)才有这样的权限。使用权限 : **root**。

命令语法：

chown [选项] user[:group] file...

常用选项:

-c : 显示更改的部分的信息；

-f : 忽略错误信息；

-R : 处理指定目录以及其子目录下的所有文件。

参数说明

user : 新的文件拥有者的使用者 ID。

group : 新的文件拥有者的使用者组(group)。

flie：文件。

命令示例：

将文件 file1.txt 的拥有者设为omm，群体的使用者dbgrp。

chown omm:dbgrp /opt/software/openGauss/clusterconfig.xml

将目前目录下的所有文件与子目录的拥有者皆设为omm，群体的使用者dbgrp。

chown -R omm:dbgrp \*

## ls

列出文件和目录的内容。

命令语法：

ls [选项] [参数]

常用选项：

-l --以长格式显示，列出文件的详细信息，如创建者，创建时间，文件的读写权限列表等等；

-a --列出文件下所有的文件，包括以"."和".."开头的隐藏文件 （Linux下文件隐藏文件是以 .开头的，如果存在 .. 代表存在着父目录）；

-d --列出目录本身而非目录内的文件，通常要与-l一起使用；

-R --同时列出所有子目录层，与-l相似，只是不显示出文件的所有者，相当于编程中的＂递归＂实现；

-t --按照时间进行文件的排序，Time（时间）；

-s --在每个文件的后面打印出文件的大小，size（大小）；

-S --以文件的大小进行排序。

参数说明：

目录或文件。

命令示例：

以长格式列出当前目录中的文件及目录：

ls -l

## cp

复制文件或者目录。

命令语法：

cp [选项] 参数1 参数2

常用选项：

-f --如果目标文件无法打开则将其移除并重试(当 -n 选项存在时则不需再选此项)；

-n --不要覆盖已存在的文件(使前面的 -i 选项失效)；

-I --覆盖前询问(使前面的 -n 选项失效)；

-p --保持指定的属性(默认：模式，所有权，时间戳)，如果可能保持附加属性：环境、链接、xattr 等；

-R,-r --复制目录及目录内的所有项目。

参数说明：

参数1：源文件。

参数2：目标文件。

命令示例：

将home目录中的abc文件复制到opt目录下：

cp /home/abc /opt

说明：目标文件存在时，会询问是否覆盖。这是因为cp是cp -i的别名。目标文件存在时，即使加了-f标志，也还会询问是否覆盖。

## rm

删除一个目录中的一个或多个文件或目录，它也可以将某个目录及其下的所有文件及子目录均删除。对于链接文件，只是删除了链接，原有文件均保持不变。

rm是一个危险的命令，使用的时候要特别当心，否则整个系统就会毁在这个命令（比如在/（根目录）下执行rm \* rf）。所以，我们在执行rm之前最好先确认一下在哪个目录，到底要删除什么东西，操作时保持高度清醒的头脑。

命令语法：

rm [选项] 文件

常用选项：

-f --忽略不存在的文件，从不给出提示；

-r --指示rm将参数中列出的全部目录和子目录均递归地删除。

参数说明：

需要删除的文件或目录。

命令示例：

删除文件：

rm qwe

注：输入rm qwe命令后，系统会询问是否删除，输入y后就会删除文件，不想删除文件则输入n。

强制删除某个文件：

rm-rf clusterconfig.log

## cat

连接文件并在标准输出上输出。这个命令常用来显示文件内容，或者将几个文件连接起来显示，或者从标准输入读取内容并显示，它常与重定向符号配合使用。

命令语法：

cat [选项] [参数]

常用选项：

-E --在每行结束显示$；

-n –由1开始对给所有输出行编号；

-b 或 --number-nonblank：和 -n 相似，只不过对于空白行不编号；

-v --使用 ^ 和 M- 符号，除了 LFD 和 TAB 之外。

参数说明：

可操作的文件名。

命令示例:

显示testfile文件的内容：

cat textfile

把 textfile1 和 textfile2 的文档内容加上行号（空白行不加）之后将内容追加到 textfile3 文档里：

cat -b textfile1 textfile2 >> textfile3

向/etc/profile中追加内容（输入EOF表示结束追加）：

cat >>/etc/profile<<EOF

>export LD\_LIBRARY\_PATH=$packagePath/script/gspylib/clib:$LD\_LIBRARY\_PATH

>EOF

注：

EOF是end of file的缩写，表示"文字流"（stream）的结尾。"文字流"可以是文件（file），也可以是标准输入（stdin）。在Linux系统之中，EOF是当系统读取到文件结尾，所返回的一个信号值（也就是-1）。