本页不打印

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 开发/优化者 | 时间 | 审核人 | 开发类型（新开发/优化） | 更新说明 |
| 朱亮/wx883694 | 2021.05.15 |  | 新开发 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 课程编码 | 适用产品 | 产品版本 | 课程版本ISSUE |
| V1R2 |  |  | V1.0 |

修订记录

openGauss3.1.0安装

实验指导手册



华为技术有限公司

|  |
| --- |
| 版权所有 © 华为技术有限公司 2023。 保留一切权利。  非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。  商标声明  C:\Users\jwx341670\Desktop\华为标志 Huawei Logo 2018\竖版标志Vertical Version\PNG\HW_POS_RBG_Vertical-150ppi.png和其他华为商标均为华为技术有限公司的商标。  本文档提及的其他所有商标或注册商标，由各自的所有人拥有。  注意  您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。  由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。 |

|  |  |
| --- | --- |
| 华为技术有限公司 | |
| 地址： | 深圳市龙岗区坂田华为总部办公楼 邮编：518129 |
| 网址： | http://[e](http://e.huawei.com/).huawei.com |

目录

[1 实验环境介绍 3](#_Toc134432149)

[1.1 实验介绍 3](#_Toc134432150)

[1.1.1 关于本实验 3](#_Toc134432151)

[1.1.2 读者知识背景 3](#_Toc134432152)

[1.1.3 实验设备介绍 3](#_Toc134432153)

[2 openGauss编译安装及基本操作 4](#_Toc134432154)

[2.1 实验介绍 4](#_Toc134432155)

[2.1.1 关于本实验 4](#_Toc134432156)

[2.1.2 实验目的 4](#_Toc134432157)

[2.2 实验任务及步骤 4](#_Toc134432158)

[2.2.1 编译前准备 4](#_Toc134432159)

[2.2.2 数据库安装编译 6](#_Toc134432160)

[2.2.3 关卡验证 9](#_Toc134432161)

[2.3 gsql连接数据库 10](#_Toc134432162)

[2.3.1 gsql连接数据库 10](#_Toc134432163)

[2.4 关卡验证 14](#_Toc134432164)

[2.5 实验小结 14](#_Toc134432165)

[2.6 思考题 14](#_Toc134432166)

[3 数据库及对象管理 15](#_Toc134432167)

[3.1 实验介绍 15](#_Toc134432168)

[3.1.1 关于本实验 15](#_Toc134432169)

[3.1.2 实验目的 15](#_Toc134432170)

[3.2 实验内容 15](#_Toc134432171)

[3.2.1 登录数据库 15](#_Toc134432172)

[3.2.2 表空间的创建、查询、修改与删除 16](#_Toc134432173)

[3.2.3 数据库的创建、查看、修改与删除 18](#_Toc134432174)

[3.2.4 行存表、列存表、MOT表的创建、查看、修改与删除 19](#_Toc134432175)

[3.2.5 用户、角色及schema管理 21](#_Toc134432176)

[3.2.6 数据导入导出 28](#_Toc134432177)

[3.3 关卡验证 36](#_Toc134432178)

[3.4 清理实验环境 36](#_Toc134432179)

[3.5 实验小结 37](#_Toc134432180)

[3.6 思考题 37](#_Toc134432181)

[4 附录：Linux操作系统相关命令 38](#_Toc134432182)

[4.1 vi/vim 38](#_Toc134432183)

[4.2 cd 38](#_Toc134432184)

[4.3 mv 39](#_Toc134432185)

[4.4 curl 40](#_Toc134432186)

[4.5 yum 41](#_Toc134432187)

[4.6 wget 41](#_Toc134432188)

[4.7 ln 42](#_Toc134432189)

[4.8 mkdir 43](#_Toc134432190)

[4.9 chmod 43](#_Toc134432191)

[4.10 chown 44](#_Toc134432192)

[4.11 ls 45](#_Toc134432193)

[4.12 cp 45](#_Toc134432194)

[4.13 rm 46](#_Toc134432195)

[4.14 cat 47](#_Toc134432196)

[5 缩略语表 48](#_Toc134432197)

# 实验环境介绍

## 实验介绍

### 关于本实验

openGauss是关系型数据库，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。

本实验主要内容为openEuler弹性云服务上单机安装部署openGauss数据库，并进行简单的数据库相关操作。

### 读者知识背景

本课程为创新实践课，为了更好地掌握本书内容，阅读本书的读者应首先具备以下基本条件：

* 具有数据库知识背景，熟悉数据库常用基本操作（查询、插入、删除、更新等）；
* 了解基本Linux知识，熟悉常用软件如vi等；
* 熟悉华为云，会申请、释放云资源，会基本操作。

### 实验设备介绍

* 组网说明

本实验环境为华为云环境，需要购买openEuler弹性云服务。

* 设备介绍

为了满足openGauss安装部署实验需要，建议每套实验环境采用以下配置。

实验软件配套关系

|  |  |
| --- | --- |
| 软件名称 | 软件版本 |
| Linux操作系统 | openEuler 20.3 LTS |
| 工具 | Putty WinSCP TPCH |
| Python | Python 3.7.X |

# openGauss编译安装及基本操作

## 实验介绍

### 关于本实验

本实验通过第三方工具，完成openGauss的编译安装。

### 实验目的

* 掌握第三方工具库编译；
* 掌握openGauss编译安装。

## 实验任务及步骤

### 编译前准备

用root用户名，使用之前设置的密码登录ECS。



创建openGauss数据库的安装用户omm及其属组dbgrp。

[root@opengauss01 ~]# **groupadd -g 1000 dbgrp**

[root@opengauss01 ~]# **useradd -g dbgrp -u 1000 -d /home/omm omm**

修改omm用户密码。

[root@opengauss01 ~]# **passwd omm**

输入修改的omm用户密码，建议设置成复杂密码。

创建openGauss源码存放及openGauss安装路径。

[root@opengauss01 ~]# **mkdir -p /opt/software/openGauss/data**

下载第三方编译库。

社区针对 centos\_7.6\_x86\_64、openEuler20.03 LTS\_arm、openEuler20.03 LTS\_ x86\_64 三种架构及操作系统已经提供了编译好的二进制，对于这三种系统架构，可以直接使用社区提供的编译好的文件openGauss-third\_party\_binarylibs.tar.gz。

[root@opengauss01 ~]# **cd /opt/software**

[root@opengauss01 software]# **wget** [**https://opengauss.obs.cn-south-1.myhuaweicloud.com/3.1.0/binarylibs/openGauss-third\_party\_binarylibs\_openEuler\_arm.tar.gz**](https://opengauss.obs.cn-south-1.myhuaweicloud.com/3.1.0/binarylibs/openGauss-third_party_binarylibs_openEuler_arm.tar.gz)

下载时间比较长大概10分钟，请耐心等待。

解压下载好的第三方编译库，并重命名为binarylibs。

[root@opengauss01 software]# **tar -zxvf openGauss-third\_party\_binarylibs\_openEuler\_arm.tar.gz**

[root@opengauss01 software]# **mv openGauss-third\_party\_binarylibs\_openEuler\_arm binarylibs/**

下载openGauss源码。

[root@opengauss01 software]# **yum install -y git --nogpgcheck**

[root@opengauss01 software]# **git clone -b 3.1.0** [**https://gitee.com/opengauss/openGauss-server.git**](https://gitee.com/opengauss/openGauss-server.git)

使用yum安装依赖包

[root@opengauss01 src]# **yum install -y libaio-devel ncurses-devel pam-devel libffi-devel libtool libtool-devel libtool-ltdl openssl-devel bison golang flex dkms-2.6.1-5.oe1.noarch python3-devel patch glibc-devel --nogpgcheck**

替换python版本。

将python3的链接设置为python的。

[root@opengauss01 src]# **cd /usr/bin**

[root@opengauss01 bin]# **mv python python.bak**

[root@opengauss01 bin]# **ln -s python3 /usr/bin/python**

检查python的版本。

[root@opengauss01 bin]# **python -V**

返回值为：Python 3.7.4

修改/opt/software路径的用户所属组及权限。

[root@opengauss01 bin]# **chown omm:dbgrp -R /opt/software**

[root@opengauss01 bin]# **chmod 755 -R /opt/software**

### 数据库安装编译

切换omm用户。

[root@opengauss01 bin]# **su - omm**

设定omm用户的环境变量。

[omm@opengauss01 ~]$ **vi ~/.bash\_profile**

添加环境变量。

将以下环境变量内容（红色加粗部分根据实际安装环境进行设置）添加至 .bash\_profile文件的末尾。

# GaussDB Code-

export CODE\_BASE=/opt/software/openGauss

export BINARYLIBS=/opt/software/binarylibs

export GAUSSHOME=$CODE\_BASE/dest/

export PGDATA=$GAUSSHOME/data

export GCC\_PATH=$BINARYLIBS/buildtools/gcc7.3/

export CC=$GCC\_PATH/gcc/bin/gcc

export CXX=$GCC\_PATH/gcc/bin/g++

export LD\_LIBRARY\_PATH=$GAUSSHOME/lib:$GCC\_PATH/gcc/lib64:$GCC\_PATH/isl/lib:$GCC\_PATH/mpc/lib/:$GCC\_PATH/mpfr/lib/:$GCC\_PATH/gmp/lib/:$LD\_LIBRARY\_PATH

export PATH=$GAUSSHOME/bin:$GCC\_PATH/gcc/bin:$PATH

使环境变量生效。

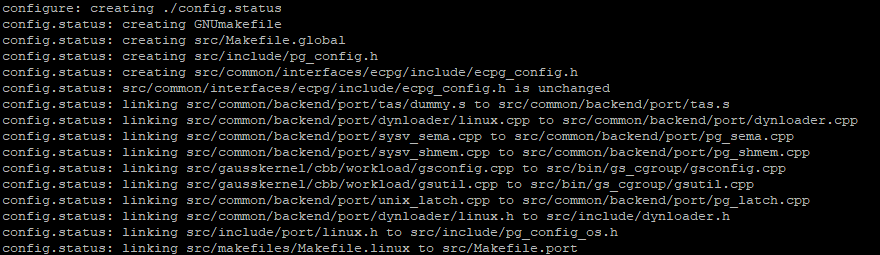
[omm@opengauss01 ~]$ **source ~/.bash\_profile**

进入openGauss源码下，生成配置文件。

[omm@opengauss01 ~]$ **cd /opt/software/openGauss-server/**

[omm@opengauss01 openGauss-server]$ **./configure --gcc-version=7.3.0 CC=g++ CFLAGS='-O0' --prefix=$GAUSSHOME --3rd=$BINARYLIBS --enable-debug --enable-cassert --enable-thread-safety --without-readline --without-zlib**

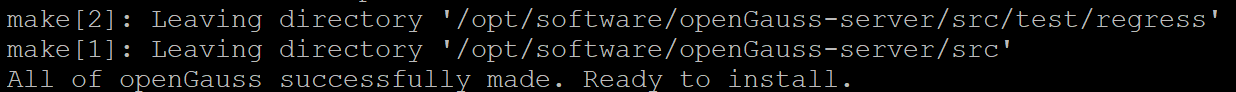
当返回如下内容时，表明成功。



执行make命令进行编译。

[omm@opengauss01 openGauss-server]$ **make -j8**

当返回All of openGauss successfully made. Ready to install.时，表明编译成功。



执行make install安装。

[omm@opengauss01 openGauss-server]$ **make install**

当返回openGauss installation complete.时，表明安装完成。

C:\Users\hwx559043\AppData\Roaming\eSpace_Desktop\UserData\hwx559043\imagefiles\E785BC09-F9C7-477E-8076-2E22F5647950.png

初始化数据库。

在初始化数据库时，需要设置数据库密码，并且要使用复杂密码，如下命令。

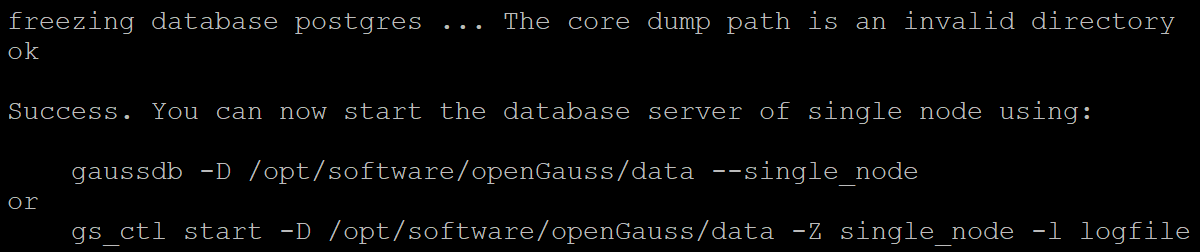
gs\_initdb -D $PGDATA --nodename=**hostname** --locale="en\_US.UTF-8" -Atrust -w **{password}**

实际使用中，将{password}部分进行替换。

例如（此处只是作为举例，建议设置为复杂密码，注意密码要用英文的单引号括起来）：

[omm@opengauss01 openGauss-server]$ **gs\_initdb -D $PGDATA --nodename=opengauss01 --locale="en\_US.UTF-8" -Atrust -w 'Huawei#!13'**

显示结果如下，表示初始化成功：



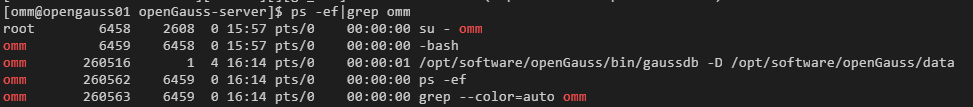
启动数据库。

[omm@opengauss01 openGauss-server]$ **gs\_ctl start -D /opt/software/openGauss/dest/data -Z single\_node -l logfile**

查看数据库是否启动成功。

[omm@opengauss01 openGauss-server]$ **ps -ef|grep omm**

显示结果如下，表示启动成功：



数据库登录。

[omm@opengauss01 openGauss-server]$ **gsql -d postgres -p 5432 -r**

修改数据库omm账号密码（可选步骤）。

openGauss=# **ALTER USER omm identified by 'Huawei@13' replace 'Huawei#!13';**

说明：ALTER USER omm identified by '新密码' replace '原密码';

查询数据库版本。

openGauss=# **select version();**

version

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

(GaussDB Kernel V500R002C00 build b2ff10be) compiled at 2021-11-22 16:34:45 commit 0 last mr debug on aarch64-unknown-linux-gnu, compiled by g++ (GCC) 7.3.0, 64-bit

(1 row)

openGauss=# **\q**

修改数据库的pg\_hba.conf文件。

[omm@opengauss01 software]$ **cd** /opt/software/openGauss/dest/data

[omm@opengauss01 data]$ cp pg\_hba.conf pg\_hba.conf.bak

[omm@opengauss01 data]$ vim pg\_hba.conf

设置 "host all all 0.0.0.0/0 sha256"

登录数据库并创建＂dboper＂用户，密码为＂**dboper@123**＂（密码可自定义），同时进行授权，并退出数据库。

[omm@opengauss01 ]$ gsql -d postgres -p 5432 -r

postgres=#CREATE USER dboper IDENTIFIED BY 'dboper@123';

CREATE ROLE

postgres=#alter user dboper sysadmin;

ALTER ROLE

postgres=# **\q**

修改数据库监听地址。

将listen\_addresses的值由原来IP地址修改成为\*号，具体操作如下

在/opt/software/openGauss/dest/data中查找**postgresql.conf**文件

[omm@opengauss01 data]$ **cd** /opt/software/openGauss/dest/data

[omm@opengauss01 data]$ **cp postgresql.conf postgresql.conf.bak**

[omm@opengauss01 data]$ **vim postgresql.conf**

设置 "listen\_addresses='\*'"

设置password\_encryption\_type = 1

修改完成后需要重启数据才能使配置生效。

### 关卡验证

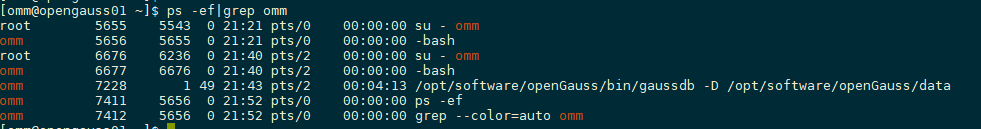
**此处需要通过截图**，来完成关卡的验证，首先需要对数据库状态进行验证。

[omm@opengauss01 openGauss-server]$ **gs\_ctl status**



其次，对数据库进程**进行截图**验证，需包含数据库服务器的主机名。

[omm@opengauss01 openGauss-server]$ **ps -ef|grep omm**



简单描述关卡一中，安装数据库所需要的步骤。完成以上信息，表明关卡一完成。

## gsql连接数据库

gsql是openGauss提供在命令行下运行的数据库连接工具，可以通过此工具连接服务器并对其进行操作和维护，除了具备操作数据库的基本功能，gsql还提供了若干高级特性，便于用户使用。

### gsql连接数据库

gsql是openGauss自带的客户端工具。使用gsql连接数据库，可以交互式地输入、编辑、执行SQL语句。

#### 连接数据库

切换到omm用户，以操作系统用户omm登录数据库主节点。

[root@opengauss01 ~]# su - omm

启动数据库服务（可选操作，安装后数据库默认会自动启动）

[omm@opengauss01 ~]# gs\_ctl start

显示如下，启动成功。



连接数据库。

执行如下命令连接数据库。

[omm@opengauss01 ~]$ gsql -d postgres -p 5432 -r

其中postgres为需要连接的数据库名称，5432为数据库主节点的端口号。请根据实际情况替换。

连接成功后，系统显示类似如下信息：

gsql ((openGauss 3.1.0 build 4e931f9a) compiled at 2022-09-29 14:19:54 commit 0 last mr )

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

Type "help" for help.

openGauss=#

omm用户是管理员用户，因此系统显示“DBNAME=#”。若使用普通用户身份登录和连接数据库，系统显示“DBNAME=>”。

“Non-SSL connection”表示未使用SSL方式连接数据库。如果需要高安全性时，请用SSL进行安全的TCP/IP连接。

退出数据库。

openGauss =# \q

#### gsql获取帮助

##### 前提条件

在omm用户。

su - omm

##### 连接数据库时，可以使用如下命令获取帮助信息

gsql --help

显示如下帮助信息：

[omm@opengauss01 ~]$ gsql --help

gsql is the openGauss interactive terminal.

Usage:

gsql [OPTION]... [DBNAME [USERNAME]]

General options:

-c, --command=COMMAND run only single command (SQL or internal) and exit

-d, --dbname=DBNAME database name to connect to (default: "omm")

-f, --file=FILENAME execute commands from file, then exit

-l, --list list available databases, then exit

-v, --set=, --variable=NAME=VALUE

set gsql variable NAME to VALUE

-V, --version output version information, then exit

-X, --no-gsqlrc do not read startup file (~/.gsqlrc)

-1 ("one"), --single-transaction

execute command file as a single transaction

-?, --help show this help, then exit

Input and output options:

-a, --echo-all echo all input from script

-e, --echo-queries echo commands sent to server

-E, --echo-hidden display queries that internal commands generate

-k, --with-key=KEY the key for decrypting the encrypted file

-L, --log-file=FILENAME send session log to file

-m, --maintenance can connect to cluster during 2-pc transaction recovery

-n, --no-libedit disable enhanced command line editing (libedit)

-o, --output=FILENAME send query results to file (or |pipe)

-q, --quiet run quietly (no messages, only query output)

-C, --enable-client-encryption enable client encryption feature

-s, --single-step single-step mode (confirm each query)

-S, --single-line single-line mode (end of line terminates SQL command)

Output format options:

-A, --no-align unaligned table output mode

-F, --field-separator=STRING

set field separator (default: "|")

-H, --html HTML table output mode

-P, --pset=VAR[=ARG] set printing option VAR to ARG (see \pset command)

-R, --record-separator=STRING

set record separator (default: newline)

-r if this parameter is set,use libedit

-t, --tuples-only print rows only

-T, --table-attr=TEXT set HTML table tag attributes (e.g., width, border)

-x, --expanded turn on expanded table output

-z, --field-separator-zero

set field separator to zero byte

-0, --record-separator-zero

set record separator to zero byte

-2, --pipeline use pipeline to pass the password, forbidden to use in terminal

must use with -c or -f

Connection options:

-h, --host=HOSTNAME database server host or socket directory (default: "/opt/huawei/wisequery/omm\_mppdb")

allow multi host IP address with comma separator in centralized cluster

-p, --port=PORT database server port (default: "5432")

-U, --username=USERNAME database user name (default: "omm")

-W, --password=PASSWORD the password of specified database user

For more information, type "\?" (for internal commands) or "\help" (for SQL

commands) from within gsql, or consult the gsql section in the openGauss documentation.

##### 连接到数据库后，可以使用如下命令获取帮助信息

使用如下命令连接数据库。

[omm@opengauss01 ~]$ gsql -d postgres -p 542 -r

输入help命令

openGauss=# help

显示如下帮助信息：

You are using gsql, the command-line interface to gaussdb.

Type: \copyright for distribution terms

\h for help with SQL commands

\? for help with gsql commands

\g or terminate with semicolon to execute query

\q to quit

查看版权信息。

openGauss=# \copyright

显示如下版本信息。

GaussDB Kernel Database Management System

Copyright (c) Huawei Technologies Co., Ltd. 2018. All rights reserved.

查看openGauss支持的所有SQL语句。

openGauss=# \h

显示如下信息：

Available help:

ABORT CREATE DATA SOURCE DROP OPERATOR

ALTER APP WORKLOAD GROUP CREATE DATABASE DROP OWNED

ALTER APP WORKLOAD GROUP MAPPING CREATE DIRECTORY DROP PACKAGE

ALTER AUDIT POLICY CREATE EXTENSION DROP PACKAGE BODY

……

查看CREATE DATABASE命令的参数可使用下面的命令。

openGauss=# \help CREATE DATABASE

显示如下帮助信息：

Command: CREATE DATABASE

Description: create a new database

Syntax:

CREATE DATABASE [ IF NOT EXISTS ] database\_name

[ [ WITH ] {[ OWNER [=] user\_name ]|

[ TEMPLATE [=] template ]|

[ ENCODING [=] encoding ]|

[ LC\_COLLATE [=] lc\_collate ]|

[ LC\_CTYPE [=] lc\_ctype ]|

[ DBCOMPATIBILITY [=] compatibility\_type ]|

[ TABLESPACE [=] tablespace\_name ]|

[ CONNECTION LIMIT [=] connlimit ]}[...] ];

查看gsql支持的元命令。

openGauss=# \?

显示如下信息：

General

\copyright show openGauss usage and distribution terms

\g [FILE] or ; execute query (and send results to file or |pipe)

\h(\help) [NAME] help on syntax of SQL commands, \* for all commands

\parallel [on [num]|off] toggle status of execute (currently off)

\q quit gsql

……

退出数据库

openGauss=# \q

## 关卡验证

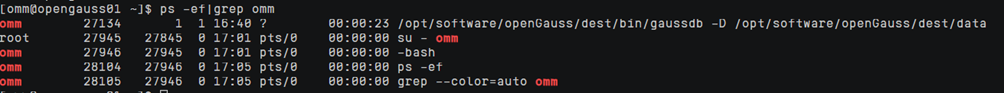
此处需要通过截图，来完成关卡的验证，首先需要对数据库状态进行验证。

[omm@opengauss01 ~]$ gs\_ctl status



其次，使用gsql连接数据库并进行截图验证，需包含数据库的系统信息。

[omm@opengauss01 ~]$ gsql -d postgres -p 5432 -r



简单描述关卡一中，安装数据库所需要的步骤。完成以上信息，表明关卡一完成。

数据库安装步骤：

（1）购买弹性云服务器

（2）修改操作系统设置

（3）安装openGauss数据库：包括下载数据库安装包，创建XML配置文件，初始化安装环境，执行安装，安装生成的目录

## 实验小结

本实验介绍了如何在openEuler弹性云服务器上安装openGauss数据库3.1.0版本，并使用客户端工具gsql进行数据库的连接。

## 思考题

gsql是openGauss提供在命令行下运行的数据库连接工具，可以使用gsql进行哪些数据库操作？

参考答案：连接数据库、推出数据库、列出数据库集簇中所有数据库的名称、所有者、字符集编码以及使用权限等。

为什么需要通过源码编译，完成数据库的安装？简单描述安装数据库所需要的步骤。

答案：源码编译适用于不同的平台，如ARM架构和X86，在底层的汇编代码上就存在着差异，源码编译可以避免这种差异，同时源码编译时，可以根据具体的需求，选择不同的组件进行编译安装。

步骤请参考PPT内容。

# 数据库及对象管理

## 实验介绍

### 关于本实验

本实验主要描述表空间、数据库的创建、查询、修改、删除管理以及行存表、列存表、MOT表的创建、查询、修改和删除管理，同时介绍用户、角色及schema管理和数据导入导出的各种方式。

### 实验目的

* 掌握表空间的创建、查询、修改、删除。
* 掌握数据库的创建、查询、修改、删除。
* 掌握行存表、列存表、MOT表的创建、查询、修改、删除。
* 掌握用户、角色及schema管理。
* 掌握不同的数据导入导出方式。

## 实验内容

### 登录数据库

使用CloudShell工具登录ECS弹性云服务器

从root系统用户切换成omm数据库用户

[root@opengauss01 ~]# su - omm

登录数据库（注意实验手册中数据库端口为5432，请确认实际端口）

[omm@opengauss01 ~]$ gsql -d postgres -p 5432 -r

gsql ((openGauss 3.1.0 build 4e931f9a) compiled at 2022-09-29 14:19:54 commit 0 last mr )

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

Type "help" for help.

openGauss=#

### 表空间的创建、查询、修改与删除

通过使用表空间，管理员可以控制一个数据库安装的磁盘布局。这样有以下优点：

1. 表空间允许管理员根据数据库对象的使用模式安排数据位置，从而提高性能。一个频繁使用的索引可以放在性能稳定且运算速度较快的磁盘上，比如一种固态设备。一个存储归档的数据，很少使用的或者对性能要求不高的表可以存储在一个运算速度较慢的磁盘上。

2. 管理员通过表空间可以设置占用的磁盘空间。用以在和其他数据共用分区的时候，防止表空间占用相同分区上的其他空间。

3. 表空间可以控制数据库数据占用的磁盘空间，当表空间所在磁盘的使用率达到90%时，数据库将被设置为只读模式，当磁盘使用率降到90%以下时，数据库将恢复到读写模式。

4. 表空间对应于一个文件系统目录，用户需要对该目录拥有读写权限。

5. 使用表空间配额管理会使性能有30%左右的影响，MAXSIZE指定每个数据库节点的配额大小，误差范围在500MB以内。请根据实际的情况确认是否需要设置表空间的最大值。

创建表空间。

openGauss=# create tablespace test\_tbs LOCATION '/opt/software/opengauss/tablespace/test\_tbs1';

CREATE TABLESPACE

当结果显示为“CREATE TABLESPACE”时，表示创建成功。

其中，test\_tbs为新创建的表空间，/opt/opengauss/tablespace/test\_tbs1为用户拥有读写权限的空目录，也是数据文件存放的目录。

创建用户，并授予test\_tbs表空间的访问权限。

openGauss=# create user jack IDENTIFIED BY 'xxxxxxxx';

CREATE ROLE

openGauss=# GRANT CREATE ON TABLESPACE test\_tbs TO jack;

GRANT

其中'xxxxxxxx'为用户密码，需自定义设置并符合密码复杂度要求。授予完成后，用户jack就具有了在test\_tbs下创建数据库对象的权限。

在test\_tbs表空间下创建表。

方法一：

openGauss=# create table test (id int) tablespace test\_tbs;

CREATE TABLE

方法二：

openGauss=# SET default\_tablespace = test\_tbs;

SET

openGauss=# create table test1 (id int);

CREATE TABLE

查询表空间。

方法一：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。

openGauss=# SELECT spcname FROM pg\_tablespace;

spcname

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

pg\_default

pg\_global

test\_tbs

(3 rows)

方法二：使用gsql程序的元命令查询表空间。

openGauss=# \db

List of tablespaces

Name | Owner | Location

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

pg\_default | omm |

pg\_global | omm |

test\_tbs | omm | /opt/opengauss/tablespace/test\_tbs1

(3 rows)

查询表空间使用率。

查询表空间的当前使用情况，其中得到的结果表示表空间的大小，单位为字节。

openGauss=# SELECT PG\_TABLESPACE\_SIZE('test\_tbs');

pg\_tablespace\_size

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

8192

(1 row)

计算表空间使用率。

表空间使用率=PG\_TABLESPACE\_SIZE/表空间所在目录的磁盘大小。

修改表空间。

通过以下命令重命名表空间。

openGauss=# ALTER TABLESPACE test\_tbs RENAME TO test\_tbs1;

ALTER TABLESPACE

删除表空间。

通过以下命令删除表空间。用户必须是表空间的owner或者系统管理员才能删除表空间。

openGauss=# drop tablespace test\_tbs1;

ERROR: tablespace "test\_tbs1" is not empty

发现有报错，提示表空间不为空。此时需要先删除在表空间内创建的对象，再删除表空间。

openGauss=# drop table test;

DROP TABLE

openGauss=# drop table test1;

DROP TABLE

openGauss=# drop tablespace test\_tbs1;

DROP TABLESPACE

### 数据库的创建、查看、修改与删除

用户必须拥有数据库创建的权限或者是数据库的系统管理员权限才能创建数据库。

创建两个新的表空间db\_tbs，db\_tbs1。

openGauss=# CREATE TABLESPACE db\_tbs LOCATION '/opt/software/opengauss/tablespace/db\_tbs1';

openGauss=# CREATE TABLESPACE db\_tbs1 LOCATION '/opt/software/opengauss/tablespace/db\_tbs2';

CREATE TABLESPACE

CREATE TABLESPACE

在db\_tbs表空间内新建数据库testdb。

openGauss=# CREATE DATABASE testdb with TABLESPACE = db\_tbs;

CREATE DATABASE

查看数据库。

方式一：通过系统表pg\_database查询数据库列表。

openGauss=# SELECT datname FROM pg\_database;

datname

-----------

template1

testdb

template0

postgres

(4 rows)

方式二：使用\l元命令查看数据库系统的数据库列表。

openGauss=# \l

List of databases

Name | Owner | Encoding | Collate | Ctype | Access privileges

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

postgres | omm | UTF8 | C | C |

template0 | omm | UTF8 | C | C | =c/omm +

| | | | | omm=CTc/omm

template1 | omm | UTF8 | C | C | =c/omm +

| | | | | omm=CTc/omm

testdb | omm | UTF8 | C | C |

(4 rows)

修改数据库。

修改数据库设置默认的模式搜索路径。

openGauss=# ALTER DATABASE testdb SET search\_path TO pa\_catalog,public;

ALTER DATABASE

修改数据库表空间。

openGauss=# ALTER DATABASE testdb SET tablespace db\_tbs1;

ALTER DATABASE

重命名数据库。

openGauss=# ALTER DATABASE testdb RENAME TO testdb1;

ALTER DATABASE

删除数据库。

openGauss=# DROP DATABASE testdb1;

DROP DATABASE

### 行存表、列存表、查看、修改与删除

行存表的适用场景：

点查询（返回记录少，基于索引的简单查询）；

增、删、改操作较多的场景。

列存表的适用场景：

统计分析类查询（关联、分组操作较多的场景）；

即席查询（查询条件不确定，行存表扫描难以使用索引）。

创建行存表。默认就是行存表，即WITH (ORIENTATION = ROW)可省略。

openGauss=# CREATE TABLE PART

(P\_PARTKEY BIGINT NOT NULL,P\_NAME VARCHAR(55) NOT NULL,P\_MFGR CHAR(25) NOT NULL,P\_BRAND CHAR(10) NOT NULL,P\_TYPE VARCHAR(25) NOT NULL,P\_SIZE BIGINT NOT NULL,P\_CONTAINER CHAR(10) NOT NULL,P\_RETAILPRICE DECIMAL(15,2) NOT NULL,P\_COMMENT VARCHAR(23) NOT NULL)

WITH (ORIENTATION = ROW);

创建列存表。

openGauss=# CREATE TABLE PART1

(P\_PARTKEY BIGINT NOT NULL,P\_NAME VARCHAR(55) NOT NULL,P\_MFGR CHAR(25) NOT NULL,P\_BRAND CHAR(10) NOT NULL,P\_TYPE VARCHAR(25) NOT NULL,P\_SIZE BIGINT NOT NULL,P\_CONTAINER CHAR(10) NOT NULL,P\_RETAILPRICE DECIMAL(15,2) NOT NULL,P\_COMMENT VARCHAR(23) NOT NULL

)

WITH (ORIENTATION = COLUMN);

查看表信息。

使用\d元命令查看表列表。

openGauss=# \d

List of relations

Schema | Name | Type | Owner | Storage

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

public | part | table | omm | {orientation=row,compression=no}

public | part1 | table | omm | {orientation=column,compression=low}

(2 rows)

修改表的属性。

增加列。

openGauss=# ALTER TABLE part ADD COLUMN p\_col1 bigint;

ALTER TABLE

增加列上的默认值。

openGauss=# ALTER TABLE part ALTER COLUMN p\_col1 SET DEFAULT 1;

ALTER TABLE

删除列上的默认值。

openGauss=# ALTER TABLE part ALTER COLUMN p\_col1 drop DEFAULT;

ALTER TABLE

修改字段的数据类型。

openGauss=# ALTER TABLE part MODIFY p\_col1 INT;

ALTER TABLE

修改列的名称。

openGauss=# ALTER TABLE part RENAME p\_col1 to p\_col;

ALTER TABLE

删除列。

openGauss=# ALTER TABLE part DROP COLUMN p\_col;

ALTER TABLE

删除表。

openGauss=# DROP TABLE PART;

DROP TABLE

openGauss=# DROP TABLE PART1;

DROP TABLE

### 用户、角色及schema管理

#### 用户管理

通过CREATE USER创建的用户，默认具有LOGIN权限。

通过CREATE USER创建用户的同时系统会在执行该命令的数据库中，为该用户创建一个同名的SCHEMA。其他数据库中，则不自动创建同名的SCHEMA；

用户可使用CREATE SCHEMA命令，会分别在其他数据库中，为该用户创建同名SCHEMA。

连接数据库后，进入SQL命令界面。创建用户jim，登录密码为Bigdata@123。

openGauss=# CREATE USER jim PASSWORD 'Bigdata@123';

CREATE ROLE

密码规则如下：

密码默认不少于8个字符；

不能与用户名及用户名倒序相同；

至少包含大写字母（A-Z），小写字母（a-z），数字（0-9），非字母数字字符（限定为~!@#$%^&\*()-\_=+\|[{}];:,<.>/?）四类字符中的三类字符；

创建用户时，应当使用双引号或单引号将用户密码括起来。

查看用户列表。

openGauss=# SELECT \* FROM pg\_user;

usename | usesysid | usecreatedb | usesuper | usecatupd | userepl | passwd | valbegin | valuntil | r

espool | parent | spacelimit | useconfig | nodegroup | tempspacelimit | spillspacelimit | usemonitorad

min | useoperatoradmin | usepolicyadmin

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

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

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

omm | 10 | t | t | t | t | \*\*\*\*\*\*\*\* | | | def

ault\_pool | 0 | | | | | | t

| t | t

jack | 16385 | f | f | f | f | \*\*\*\*\*\*\*\* | | | def

ault\_pool | 0 | | | | | | f

| f | f

jim | 16427 | f | f | f | f | \*\*\*\*\*\*\*\* | | | def

ault\_pool | 0 | | | | | | f

| f | f

(3 rows)

创建有“创建数据库”权限的用户，则需要加CREATEDB关键字。

openGauss=# CREATE USER dim CREATEDB PASSWORD 'Bigdata@123';

CREATE ROLE

将用户jim的登录密码由Bigdata@123修改为Abcd@123。

openGauss=# ALTER USER jim IDENTIFIED BY 'Abcd@123' REPLACE 'Bigdata@123';

ALTER ROLE

为用户jim追加有创建角色的CREATEROLE权限。

openGauss=# ALTER USER jim CREATEROLE;

ALTER ROLE

锁定jim帐户。

openGauss=# ALTER USER jim ACCOUNT LOCK;

ALTER ROLE

解锁jim帐户。

openGauss=# ALTER USER jim ACCOUNT UNLOCK;

ALTER ROLE

删除用户。

openGauss=# DROP USER jim CASCADE;

DROP ROLE

openGauss=# DROP USER dim CASCADE;

DROP ROLE

#### 角色管理

角色是拥有数据库对象和权限的实体；

在不同的环境中角色可以认为是一个用户，一个组或者兼顾两者；

在数据库中添加一个新角色，角色无登录权限；

创建角色的用户必须具备CREATE ROLE的权限或者是系统管理员。

创建一个角色，名为manager，密码为Bigdata@123。

openGauss=# CREATE ROLE manager IDENTIFIED BY 'Bigdata@123';

CREATE ROLE

创建一个角色，从2020年7月1日开始生效，到2020年12月1日失效。

openGauss=# CREATE ROLE miriam WITH LOGIN PASSWORD 'Bigdata@123' VALID BEGIN '2020-07-01' VALID UNTIL '2020-12-01';

CREATE ROLE

修改角色manager为系统管理员。

openGauss=# ALTER ROLE manager SYSADMIN;

ALTER ROLE

删除角色manager。

openGauss=# DROP ROLE manager;

DROP ROLE

查看角色。

openGauss=# SELECT \* FROM PG\_ROLES;

rolname | rolsuper | rolinherit | rolcreaterole | rolcreatedb | rolcatupdate | rolcanlo

gin | rolreplication | rolauditadmin | rolsystemadmin | rolconnlimit | rolpassword | rolvalidbegin

| rolvaliduntil | rolrespool | rolparentid | roltabspace | rolconfig | oid | roluseft | r

olkind | nodegroup | roltempspace | rolspillspace | rolmonitoradmin | roloperatoradmin | rolpolicyadmin

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

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

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

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

omm | t | t | t | t | t | t

| t | t | t | -1 | \*\*\*\*\*\*\*\* |

| | default\_pool | 0 | | | 10 | t | n

| | | | t | t | t

gs\_role\_directory\_drop | f | t | f | f | f | f

| f | f | f | -1 | \*\*\*\*\*\*\*\* |

| | default\_pool | 0 | | | 1059 | f | n

| | | | f | f | f

gs\_role\_directory\_create | f | t | f | f | f | f

| f | f | f | -1 | \*\*\*\*\*\*\*\* |

| | default\_pool | 0 | | | 1056 | f | n

| | | | f | f | f

gs\_role\_pldebugger | f | t | f | f | f | f

| f | f | f | -1 | \*\*\*\*\*\*\*\* |

| | default\_pool | 0 | | | 1055 | f | n

| | | | f | f | f

gs\_role\_account\_lock | f | t | f | f | f | f

| f | f | f | -1 | \*\*\*\*\*\*\*\* |

| | default\_pool | 0 | | | 1048 | f | n

| | | | f | f | f

gs\_role\_replication | f | t | f | f | f | f

| f | f | f | -1 | \*\*\*\*\*\*\*\* |

| | default\_pool | 0 | | | 1047 | f | n

| | | | f | f | f

gs\_role\_tablespace | f | t | f | f | f | f

| f | f | f | -1 | \*\*\*\*\*\*\*\* |

| | default\_pool | 0 | | | 1046 | f | n

| | | | f | f | f

gs\_role\_signal\_backend | f | t | f | f | f | f

| f | f | f | -1 | \*\*\*\*\*\*\*\* |

| | default\_pool | 0 | | | 1045 | f | n

| | | | f | f | f

gs\_role\_copy\_files | f | t | f | f | f | f

| f | f | f | -1 | \*\*\*\*\*\*\*\* |

| | default\_pool | 0 | | | 1044 | f | n

| | | | f | f | f

miriam | f | t | f | f | f | t

| f | f | f | -1 | \*\*\*\*\*\*\*\* | 2020-07-01 00:00:00+

08 | 2020-12-01 00:00:00+08 | default\_pool | 0 | | | 16439 | f | n

| | | | f | f | f

jack | f | t | f | f | f | t

| f | f | f | -1 | \*\*\*\*\*\*\*\* |

| | default\_pool | 0 | | | 16385 | f | n

| | | | f | f | f

(11 rows)

删除角色miriam。

openGauss=# DROP ROLE miriam;

DROP ROLE

#### Schema管理

Schema又称作模式。

通过管理Schema，允许多个用户使用同一数据库而不相互干扰；

每个数据库包含一个或多个Schema；

在数据库创建用户时，系统会自动帮助用户创建一个同名Schema。

创建模式ds。

openGauss=# CREATE SCHEMA ds;

CREATE SCHEMA

将当前模式ds更名为ds\_new。

openGauss=# ALTER SCHEMA ds RENAME TO ds\_new;

ALTER SCHEMA

创建用户dim。

openGauss=# CREATE USER dim PASSWORD 'Bigdata@123';

CREATE ROLE

将DS\_NEW的所有者修改为jack。

openGauss=# ALTER SCHEMA ds\_new OWNER TO jack;

ALTER SCHEMA

查看Schema所有者。

openGauss=# SELECT s.nspname,u.usename AS nspowner FROM pg\_namespace s, pg\_user u WHERE s.nspowner = u.usesysid;

nspname | nspowner

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

pg\_toast | omm

cstore | omm

pkg\_service | omm

dbe\_perf | omm

snapshot | omm

blockchain | omm

pg\_catalog | omm

public | omm

sqladvisor | omm

dbe\_pldebugger | omm

dbe\_pldeveloper | omm

dbe\_sql\_util | omm

information\_schema | omm

db4ai | omm

jack | jack

dim | dim

ds\_new | jack

(17 rows)

删除用户jack和模式ds\_new。

openGauss=# DROP SCHEMA ds\_new;

DROP SCHEMA

openGauss=# DROP USER jack;

DROP ROLE

#### 将系统权限授权给用户或者角色

创建名为joe的用户，并将sysadmin权限授权给他。

openGauss=# CREATE USER joe PASSWORD 'Bigdata@123';

CREATE ROLE

openGauss=# ALTER USER joe with sysadmin;

ALTER ROLE

#### 将数据库对象授权给角色或用户

撤销joe用户的sysadmin权限，然后创建tpcds模式，并给tpcds模式下创建一张reason表。

openGauss=# ALTER USER joe with nosysadmin;

ALTER ROLE

openGauss=# CREATE SCHEMA tpcds;

CREATE SCHEMA

openGauss=# CREATE TABLE tpcds.reason (r\_reason\_sk INTEGER NOT NULL,r\_reason\_id CHAR(16) NOT NULL,r\_reason\_desc VARCHAR(20)

);

CREATE TABLE

将模式tpcds的使用权限和表tpcds.reason的所有权限授权给用户joe。

openGauss=# GRANT USAGE ON SCHEMA tpcds TO joe;

GRANT

openGauss=# GRANT ALL PRIVILEGES ON tpcds.reason TO joe;

GRANT

授权成功后，joe用户就拥有了tpcds.reason表的所有权限，包括增删改查等权限。

将tpcds.reason表中r\_reason\_sk、r\_reason\_id、r\_reason\_desc列的查询权限，r\_reason\_desc的更新权限授权给joe。

openGauss=# G**RANT select (r\_reason\_sk,r\_reason\_id,r\_reason\_desc),update (r\_reason\_desc) ON tpcds.reason TO joe;**

**GRANT**

将数据库postgres的连接权限授权给用户joe，并给予其在postgres中创建schema的权限，而且允许joe将此权限授权给其他用户。

openGauss=# GRANT create,connect on database postgres TO joe WITH GRANT OPTION;

GRANT

创建角色tpcds\_manager，将模式tpcds的访问权限授权给角色tpcds\_manager，并授予该角色在tpcds下创建对象的权限，不允许该角色中的用户将权限授权给他人。

openGauss=# CREATE ROLE tpcds\_manager PASSWORD 'Bigdata@123';

CREATE ROLE

openGauss=# GRANT USAGE,CREATE ON SCHEMA tpcds TO tpcds\_manager;

GRANT

#### 将用户或者角色的权限授权给其他用户或角色

创建角色manager，将joe的权限授权给manager，并允许该角色将权限授权给其他人。

openGauss=# CREATE ROLE manager PASSWORD 'Bigdata@123';

CREATE ROLE

openGauss=# GRANT joe TO manager WITH ADMIN OPTION;

GRANT ROLE

创建用户senior\_manager，将用户manager的权限授权给该用户。

openGauss=# CREATE USER senior\_manager PASSWORD 'Bigdata@123';

CREATE ROLE

openGauss=# GRANT manager TO senior\_manager;

GRANT ROLE

#### 权限回收

撤销权限，并清理用户

openGauss=# DROP USER manager;

DROP ROLE

openGauss=# REVOKE ALL PRIVILEGES ON tpcds.reason FROM joe;

REVOKE

openGauss=# REVOKE ALL PRIVILEGES ON SCHEMA tpcds FROM joe;

REVOKE

openGauss=# REVOKE USAGE,CREATE ON SCHEMA tpcds FROM tpcds\_manager;

REVOKE

openGauss=# DROP ROLE tpcds\_manager;

DROP ROLE

openGauss=# DROP USER senior\_manager;

DROP ROLE

openGauss=# DROP USER joe CASCADE;

DROP ROLE

openGauss=# \q

### 数据导入导出

gs\_dump是openGauss用于导出数据库相关信息的工具，用户可以自定义导出一个数据库或其中的对象（模式、表、视图等）。支持导出的数据库可以是默认数据库postgres，也可以是自定义数据库。

#### 创建备份目录

使用omm用户，创建备份目录。

[omm@opengauss01 ~]$ mkdir ./gaussdb/backup /

[omm@opengauss01 ~]$ cd ./gaussdb/backup

#### 创建表并插入数据

登录数据库。

[omm@opengauss01 backup]$ gsql -d postgres -p 5432 -r

创建customer\_t1表。

openGauss=# drop table if exists customer\_t1;

NOTICE: table "customer\_t1" does not exist, skipping

DROP TABLE

openGauss=# CREATE TABLE customer\_t1 (

c\_customer\_sk integer,

c\_customer\_id char(5),

c\_first\_name char(6),

c\_last\_name char(8));

CREATE TABLE

向customer\_t1表中插入数据。

openGauss=# INSERT INTO customer\_t1 (c\_customer\_sk, c\_customer\_id, c\_first\_name) VALUES

(3769, 'hello', DEFAULT) ,

(6885, 'maps', 'Joes'),

(4321, 'tpcds', 'Lily'),

(9527, 'world', 'James');

INSERT 0 4

查看表customer\_t1数据。

openGauss=# select \* from customer\_t1;

c\_customer\_sk | c\_customer\_id | c\_first\_name | c\_last\_name

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

3769 | hello | |

6885 | maps | Joes |

4321 | tpcds | Lily |

9527 | world | James |

(4 rows)

创建joe用户。

openGauss=# drop user if exists joe;

NOTICE: role "joe" does not exist, skipping

DROP ROLE

openGauss=# CREATE USER joe WITH PASSWORD "Bigdata@123";

CREATE ROLE

在joe用户下创建表mytable。

openGauss=# \q

[omm@opengauss01 ~]$ gsql -d postgres -U joe -W Bigdata@123 -p 5432 -r

gsql ((openGauss 3.1.0 build 4e931f9a) compiled at 2022-09-29 14:19:54 commit 0 last mr )

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

Type "help" for help.

openGauss=> drop table if exists mytable;

NOTICE: table "mytable" does not exist, skipping

DROP TABLE

openGauss=> CREATE TABLE mytable (firstcol int);

CREATE TABLE

向mytable表插入数据。

openGauss=> INSERT INTO mytable values (100);

INSERT 0 1

查看表mytable文件。

openGauss=> SELECT \* from mytable;

firstcol

----------

100

(1 row)

查看当前表信息。

openGauss=> \d

List of relations

Schema | Name | Type | Owner | Storage

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

joe | mytable | table | joe | {orientation=row,compression=no}

public | customer\_t1 | table | | {orientation=row,compression=no}

(2 rows)

#### copy数据导出

切换到默认omm用户下。

openGauss=> \c - omm

使用copy命令进行导出。

openGauss=# copy customer\_t1 to '/home/omm/gaussdb/backup/copy\_cost.txt' delimiter '^';

COPY 4

openGauss=# \q

查看导出文件。

[omm@opengauss01 ~]$ cd /home/omm/gaussdb/backup

[omm@opengauss01 backup]$ ll

total 4.0K

-rw------- 1 omm dbgrp 80 Jan 12 22:59 copy\_cost.txt

[omm@opengauss01 backup]$ more copy\_cost.txt

3769^hello^\N^\N

6885^maps ^Joes ^\N

4321^tpcds^Lily ^\N

9527^world^James ^\N

#### gs\_dump数据导出

使用gs\_dump命令将postgres数据库导出，导出为纯文本格式。

gs\_dump部分重要指令解释：

-f：将导出文件发送至指定目录文件夹。

-F：选择导出文件格式。-F参数值如下：

p：纯文本格式

c：自定义归档

d：目录归档格式

t：tar归档格式

-n：只导出与模式名称匹配的模式，此选项包括模式本身和所有它包含的对象。

-t：指定导出的表（或视图、序列、外表），可以使用多个-t选项来选择多个表，也可以使用通配符指定多个表对象。

-T：不转储的表（或视图、或序列、或外表）对象列表，可以使用多个-T选项来选择多个表，也可以使用通配符指定多个表对象。

[omm@opengauss01 backup]$ gs\_dump -U omm -W Bigdata@123 -f /home/omm/gaussdb/backup/gsdump\_post.sql -p 5432 postgres -F p

gs\_dump[port='5432'][postgres][2023-01-12 23:01:56]: The total objects number is 436.

gs\_dump[port='5432'][postgres][2023-01-12 23:01:56]: [100.00%] 436 objects have been dumped.

gs\_dump[port='5432'][postgres][2023-01-12 23:01:56]: dump database postgres successfully

gs\_dump[port='5432'][postgres][2023-01-12 23:01:56]: total time: 1637 ms

查看导出文件。

[omm@opengauss01 backup]$ cd /home/omm/gaussdb/backup/

[omm@opengauss01 backup]$ ll

total 8.0K

-rw------- 1 omm dbgrp 80 Jan 12 22:59 copy\_cost.txt

-rw------- 1 omm dbgrp 2.9K Jan 12 23:01 gsdump\_post.sql

[omm@opengauss01 backup]$ cat gsdump\_post.sql |grep -v ^$

--

-- openGauss database dump

--

SET statement\_timeout = 0;

SET xmloption = content;

SET client\_encoding = 'UTF8';

SET standard\_conforming\_strings = on;

SET check\_function\_bodies = false;

SET client\_min\_messages = warning;

--

-- Name: postgres; Type: COMMENT; Schema: -; Owner: omm

--

COMMENT ON DATABASE postgres IS 'default administrative connection database';

--

-- Name: dim; Type: SCHEMA; Schema: -; Owner: dim

--

CREATE SCHEMA dim;

ALTER SCHEMA dim OWNER TO dim;

--

-- Name: joe; Type: SCHEMA; Schema: -; Owner: joe

--

CREATE SCHEMA joe;

ALTER SCHEMA joe OWNER TO joe;

--

-- Name: tpcds; Type: SCHEMA; Schema: -; Owner: omm

--

……

使用gs\_dump，导出postgres数据库，导出为tar格式。

[omm@opengauss01 backup]$ gs\_dump -U omm -W Bigdata@123 -f /home/omm/gaussdb/backup/gsdump\_post.tar -p 5432 postgres -F t

gs\_dump[port='5432'][postgres][2023-01-12 23:04:38]: The total objects number is 436.

gs\_dump[port='5432'][postgres][2023-01-12 23:04:38]: [100.00%] 436 objects have been dumped.

gs\_dump[port='5432'][postgres][2023-01-12 23:04:38]: dump database postgres successfully

gs\_dump[port='5432'][postgres][2023-01-12 23:04:38]: total time: 1668 ms

查看导出文件。

[omm@opengauss01 backup]$ ll

total 24K

-rw------- 1 omm dbgrp 80 Jan 12 22:59 copy\_cost.txt

-rw------- 1 omm dbgrp 2.9K Jan 12 23:01 gsdump\_post.sql

-rw------- 1 omm dbgrp 13K Jan 12 23:04 gsdump\_post.tar

使用gs\_dump导出依赖于表的视图。

连接数据库，并创建依赖于表customer\_t1的视图vw\_customer\_t1，后退出。

[omm@opengauss01 backup]$ gsql -d postgres -p 5432 -r

openGauss=# create view vw\_customer\_t1 as select \* from customer\_t1 limit 2;

CREATE VIEW

openGauss=# \q

导出依赖于表customer\_t1的视图vw\_customer\_t1。

[omm@opengauss01 backup]$ gs\_dump -s -p 5432 postgres -t PUBLIC.customer\_t1 --include-depend-objs --exclude-self -f /home/omm/gaussdb/backup/view\_cust.sql -F p

gs\_dump[port='5432'][postgres][2023-01-12 23:05:43]: The total objects number is 421.

gs\_dump[port='5432'][postgres][2023-01-12 23:05:43]: [100.00%] 421 objects have been dumped.

gs\_dump[port='5432'][postgres][2023-01-12 23:05:43]: dump database postgres successfully

gs\_dump[port='5432'][postgres][2023-01-12 23:05:43]: total time: 1621 ms

查看导出文件。

[omm@opengauss01 backup]$ cd /home/omm/gaussdb/backup/

[omm@opengauss01 backup]$ ll

total 28K

-rw------- 1 omm dbgrp 80 Jan 12 22:59 copy\_cost.txt

-rw------- 1 omm dbgrp 2.9K Jan 12 23:01 gsdump\_post.sql

-rw------- 1 omm dbgrp 13K Jan 12 23:04 gsdump\_post.tar

-rw------- 1 omm dbgrp 607 Jan 12 23:05 view\_cust.sql

#### gs\_dumpall数据导出。

gs\_dumpall是openGauss用于导出所有数据库相关信息工具，它可以导出openGauss数据库的所有数据，包括默认数据库postgres的数据、自定义数据库的数据、以及openGauss所有数据库公共的全局对象。

gs\_dumpall在导出openGauss所有数据库时分为两部分：

* gs\_dumpall自身对所有数据库公共的全局对象进行导出，包括有关数据库用户和组，表空间以及属性（例如，适用于数据库整体的访问权限）信息。
* gs\_dumpall通过调用gs\_dump来完成openGauss中各数据库的SQL脚本文件导出，该脚本文件包含将数据库恢复为其保存时的状态所需要的全部SQL语句。

以上两部分导出的结果为纯文本格式的SQL脚本文件，使用gsql运行该脚本文件可以恢复openGauss数据库。

使用gs\_dumpall一次导出openGauss的所有数据库。

[omm@opengauss01 backup]$ gs\_dumpall -f /home/omm/gaussdb/backup/gsdumpall.sql -p 5432

gs\_dump[port='5432'][dbname='postgres'][2023-01-12 23:06:27]: The total objects number is 438.

gs\_dump[port='5432'][dbname='postgres'][2023-01-12 23:06:27]: [100.00%] 438 objects have been dumped.

gs\_dump[port='5432'][dbname='postgres'][2023-01-12 23:06:27]: dump database dbname='postgres' successfully

gs\_dump[port='5432'][dbname='postgres'][2023-01-12 23:06:27]: total time: 1725 ms

gs\_dumpall[port='5432'][2023-01-12 23:06:27]: dumpall operation successful

gs\_dumpall[port='5432'][2023-01-12 23:06:27]: total time: 1766 ms

查看导出文件。

[omm@opengauss01 backup]$ cd /home/omm/gaussdb/backup/

[omm@opengauss01 backup]$ ll

total 36K

-rw------- 1 omm dbgrp 80 Jan 12 22:59 copy\_cost.txt

-rw------- 1 omm dbgrp 6.4K Jan 12 23:06 gsdumpall.sql

-rw------- 1 omm dbgrp 2.9K Jan 12 23:01 gsdump\_post.sql

-rw------- 1 omm dbgrp 13K Jan 12 23:04 gsdump\_post.tar

-rw------- 1 omm dbgrp 607 Jan 12 23:05 view\_cust.sql

#### gsql数据导入

对于.sql文件，可使用gsql直接导入。

删除之前导出的表。

[omm@opengauss01 backup]$ gsql -p 5432 postgres -r

gsql ((openGauss 3.1.0 build 4e931f9a) compiled at 2022-09-29 14:19:54 commit 0 last mr )

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

Type "help" for help.

openGauss=# drop table joe.mytable;

DROP TABLE

openGauss=# drop table public.customer\_t1 cascade;

NOTICE: drop cascades to view vw\_customer\_t1

DROP TABLE

openGauss=# \q

使用gsql将导出的表恢复。（出现ERROR: schema "joe" already exists报错可忽略）

[omm@opengauss01 backup]$ gsql -p 5432 postgres -r -f /home/omm/gaussdb/backup/gsdump\_post.sql

查看恢复后的表。

[omm@opengauss01 backup]$ gsql -p 5432 postgres -r

gsql ((openGauss 3.1.0 build 4e931f9a) compiled at 2022-09-29 14:19:54 commit 0 last mr )

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

Type "help" for help.

openGauss=# select \* from joe.mytable;

firstcol

----------

100

(1 row)

openGauss=# select \* from public.customer\_t1;

c\_customer\_sk | c\_customer\_id | c\_first\_name | c\_last\_name

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

3769 | hello | |

6885 | maps | Joes |

4321 | tpcds | Lily |

9527 | world | James |

(4 rows)

openGauss=#\q

#### copy数据导入

对于使用copy导出的数据，同样可以使用copy导入。

删除使用copy命令导出过的表。

[omm@opengauss01 backup]$ gsql -p 5432 postgres -r

gsql ((openGauss 3.1.0 build 4e931f9a) compiled at 2022-09-29 14:19:54 commit 0 last mr )

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

Type "help" for help.

openGauss=# truncate table public.customer\_t1;

TRUNCATE TABLE

使用copy命令进行导入。

openGauss=# copy customer\_t1 from '/home/omm/gaussdb/backup/copy\_cost.txt' delimiter '^';

COPY 4

查看导入的表数据。

openGauss=# select \* from customer\_t1;

c\_customer\_sk | c\_customer\_id | c\_first\_name | c\_last\_name

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

3769 | hello | |

6885 | maps | Joes |

4321 | tpcds | Lily |

9527 | world | James |

(4 rows)

openGauss=# \q

#### gs\_restore数据导入

gs\_restore是openGauss提供的针对gs\_dump导出数据的导入工具。通过此工具可由gs\_dump生成的导出文件进行导入。gs\_restore工具由操作系统用户omm执行。

删除使用gs\_dump导出的表和视图。

[omm@opengauss01gauss backup]$ gsql -p 5432 postgres -r

gsql ((openGauss 3.1.0 build 4e931f9a) compiled at 2022-09-29 14:19:54 commit 0 last mr )

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

Type "help" for help.

openGauss=# drop table joe.mytable;

DROP TABLE

openGauss=# drop table public.customer\_t1;

DROP TABLE

使用gs\_restore导入gsdump\_post.tar文件内数据到tpcc数据库。

新建tpcc数据库。

openGauss=# create database tpcc;

CREATE DATABASE

openGauss=# \q

导入gsdump\_post.tar文件内数据到tpcc数据库。

[omm@opengauss01 backup]$ gs\_restore /home/omm/gaussdb/backup/gsdump\_post.tar -p 5432 -d tpcc

start restore operation ...

table mytable complete data imported !

table customer\_t1 complete data imported !

table reason complete data imported !

Finish reading 18 SQL statements!

end restore operation ...

restore operation successful

total time: 30 ms

登录tpcc数据库查看恢复的数据。

[omm@opengauss01 backup]$ gsql -p 5432 tpcc -r

gsql ((openGauss 3.1.0 build 4e931f9a) compiled at 2022-09-29 14:19:54 commit 0 last mr )

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

Type "help" for help.

tpcc=# select \* from public.customer\_t1;

c\_customer\_sk | c\_customer\_id | c\_first\_name | c\_last\_name

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

3769 | hello | |

6885 | maps | Joes |

4321 | tpcds | Lily |

9527 | world | James |

(4 rows)

tpcc=# select \* from joe.mytable;

firstcol

----------

100

(1 row)

tpcc=#\q

## 关卡验证

通过系统表pg\_database查询数据库列表并截图。

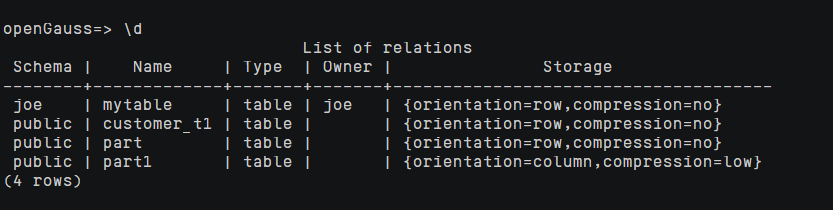
openGauss=# SELECT datname FROM pg\_database;

文本

描述已自动生成

使用元命令查看表信息，并截图。

openGauss=# \d

进行部分操作后：

进行完整操作后:

查看用户列表并截图。

openGauss=# SELECT \* FROM pg\_user;

图形用户界面

低可信度描述已自动生成

## 清理实验环境

删除创建的数据库及用户。

[omm@opengauss01 backup]$ gsql -p 5432 postgres -r

gsql ((openGauss 3.1.0 build 4e931f9a) compiled at 2022-09-29 14:19:54 commit 0 last mr )

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

Type "help" for help.

openGauss=# \l

List of databases

Name | Owner | Encoding | Collate | Ctype | Access privileges

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

postgres | omm | UTF8 | C | C | =Tc/omm +

| | | | | omm=CTc/omm

template0 | omm | UTF8 | C | C | =c/omm +

| | | | | omm=CTc/omm

template1 | omm | UTF8 | C | C | =c/omm +

| | | | | omm=CTc/omm

tpcc | omm | UTF8 | C | C |

(4 rows)

openGauss=# drop database tpcc;

DROP DATABASE

openGauss=# \du

List of roles

Role name | Attributes

| Member of

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

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

dim |

| {}

joe |

| {}

omm | Sysadmin, Create role, Create DB, Replication, Administer audit, Monitoradmin, Operatoradmin

, Policyadmin, UseFT | {}

openGauss=# drop user joe;

DROP ROLE

openGauss=#\q

## 实验小结

本实验介绍了如何登录openGauss数据库，并对openGauss数据库进行如表空间、数据库、表、用户角色等维护，并使用导入导出工具，将数据导入至openGauss数据库中，通过本节的实验，希望能够熟练掌握openGauss数据库及对象的管理。

## 思考题

1. 假设有以下操作：

a. 创建表customer\_t1，并导入4条数据；

b. 创建依赖于表customer\_t1的视图vw\_customer\_t1；

create view vw\_customer\_t1 as select \* from customer\_t1 limit 2;

此时查看视图vw\_customer\_t1，显示2条数据。

c. 导出依赖于表customer\_t1的视图vw\_customer\_t1到文件vw\_customer.sql；

d. 重命名表customer\_t1为customer\_t1\_bak；

e. 新建表customer\_t1，不插入数据；

f. 导入vw\_customer.sql；

请问此时查看视图vw\_customer，显示有几条数据？

参考答案：视图vw\_customer中无数据。

# 附录：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/3.1.0/arm/openGauss-3.1.0-openEuler-64bit-all.tar.gz

* 使用wget断点续传：

wget –c https://opengauss.obs.cn-south-1.myhuaweicloud.com/3.1.0/arm/openGauss-3.1.0-openEuler-64bit-all.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

注：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）。

# 缩略语表

|  |  |  |
| --- | --- | --- |
| 缩略语 | 英文全称 | 中文全称 |
| ECS | Elastic Cloud Server | 弹性云服务器，由CPU、内存、镜像、云硬盘组成的一种可随时获取、弹性可扩展的计算服务器。 |