

《数据库》

在虚拟机+CentOS 上安装部署

openGauss 数据库指导手册

目录

前 言	2
简介	2
内容描述	2
前置条件	2
实验环境说明	2
单机安装概览	3
1 openGauss 数据库安装	4
1.1 实验介绍	4
1.1.1 关于本实验	4
1.1.2 实验目的	4
1.2 虚拟机 VirtualBox 下载及安装	4
1.3 CentOS 7.8 镜像文件下载	5
1.4 VirtualBox 下安装 CentOS 7.8 操作系统	6
1.5 操作系统环境准备	25
1.6 安装 openGauss 数据库	28
2 数据库使用	33
2.1 前提条件	33
2.2 操作步骤	33
3 附录一：Linux 操作系统相关命令	36
4 附录二：openGauss 数据库基本操作	47

前言

简介

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。

本实验主要描述 openGauss 数据库在 CentOS 7.8 上的单机安装部署。

内容描述

本实验主要内容为在 CentOS 7.8 上安装部署 openGauss 数据库，并进行简单的数据库相关操作。

前置条件

由于本实验主要是在 CentOS 操作系统上进行 openGauss 数据库的部署，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。

实验环境说明

- 组网说明

本实验环境为虚拟机 VirtualBox 6.1.14 + CentOS 7.8 + openGauss 1.1.0。

- 设备介绍

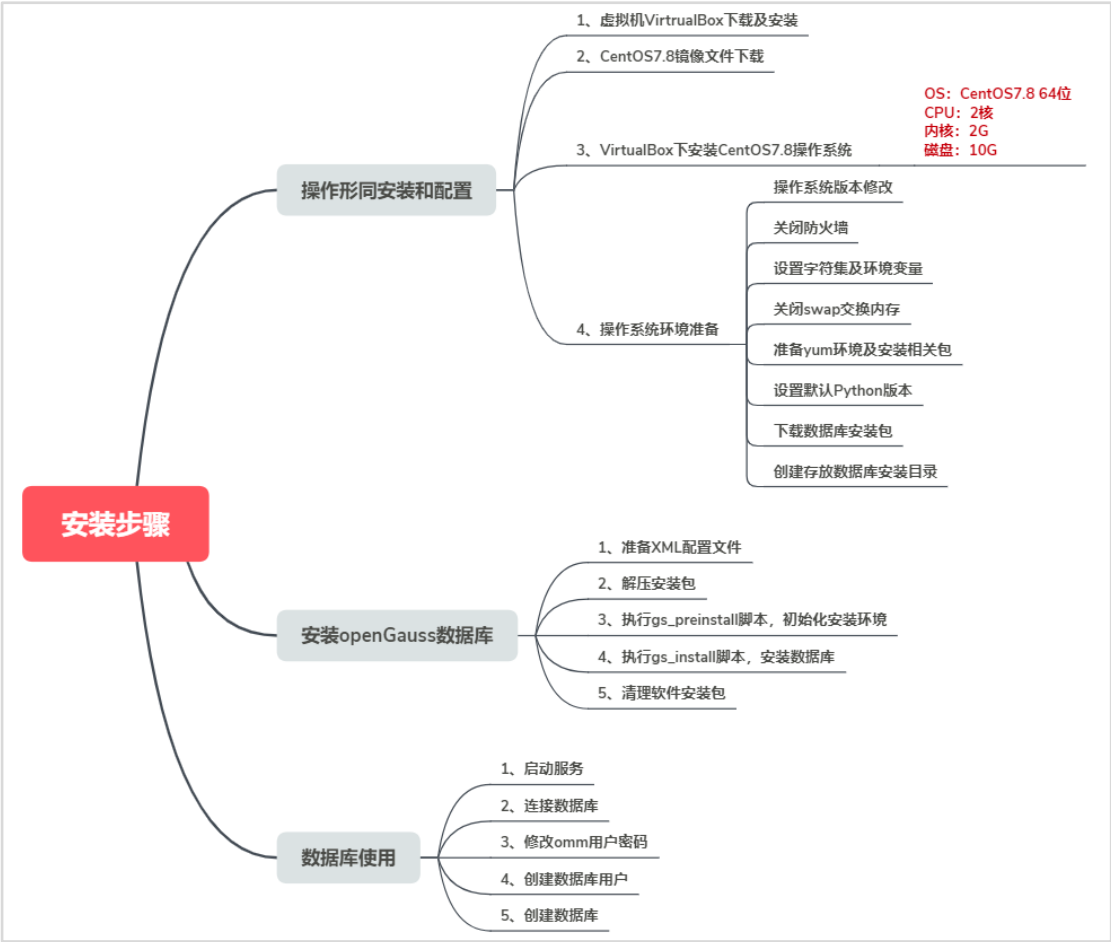
为了满足 openGauss 安装部署实验需要，建议每套实验环境采用以下配置：

设备明细表

设备名称	设备型号
Linux操作系统	CentOS 7.8
windows操作系统	win10 x86 64位

虚拟机	VirtualBox 6.1.14
Python	Python 3.6.X

单机安装概览



本实验概览图

1 openGauss 数据库安装

1.1 实验介绍

1.1.1 关于本实验

本实验主要描述 openGauss 数据库在虚拟机 VirtualBox+CentOS 上的安装配置。

1.1.2 实验目的

- 掌握虚拟机 VirtualBox 的安装配置方法；
- 掌握 openGauss 数据库安装部署方法。

1.2 虚拟机 VirtualBox 下载及安装

步骤 1 进入官方网站下载页面。

网址：<https://www.virtualbox.org/wiki/Downloads>



点击“window 主机”下载 windows 版本的 VirtualBox。

步骤 2 下载完成后，双击执行文件进行安装。

下载后，文件名为：VirtualBox-6.1.14-140239-Win.exe，双击此执行文件进行安装，安装过程中存放地址可以根据自己想法去设置下，其他所有选项都可以默认，直接按下一步就行，最后安装成功。

具体如下：



1.3 CentOS 7.8 镜像文件下载

步骤 1 进入华为开源镜像站的下载页面。

网址：https://mirrors.huaweicloud.com/centos-vault/7.8.2003/isos/x86_64/，具体如下：

/mirrors_os/centos/7.8.2003/isos/x86_64/		
File Name ↓	File Size ↓	Date ↓
Parent directory/	-	-
0_README.txt	2.4 KiB	2020-Apr-24 21:02
CentOS-7-x86_64-DVD-2003.iso	4.5 GiB	2020-Apr-22 09:05
CentOS-7-x86_64-DVD-2003.torrent	178.7 KiB	2020-Apr-24 21:38
CentOS-7-x86_64-Everything-2003.iso	9.6 GiB	2020-Apr-22 09:06
CentOS-7-x86_64-Everything-2003.torrent	383.3 KiB	2020-Apr-24 21:39
CentOS-7-x86_64-LiveGNOME-2003.iso	1.4 GiB	2020-Apr-23 06:02

步骤 2 点击“CentOS-7-x86_64-DVD-2003.iso”，进行相应 ISO 镜像文件下载。

下载的文件名如下：

名称	修改日期	类型	大小
CentOS-7-x86_64-DVD-2003.iso	2020/9/4 16:02	ISO 文件	4,669,440 KB

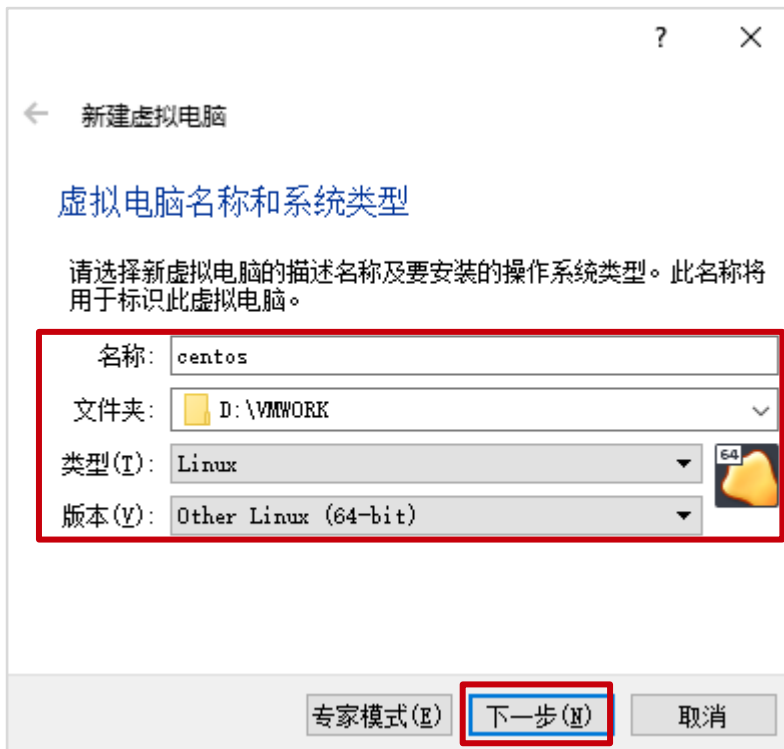
1.4 VirtualBox 下安装 CentOS 7.8 操作系统

步骤 1 新建虚拟电脑。

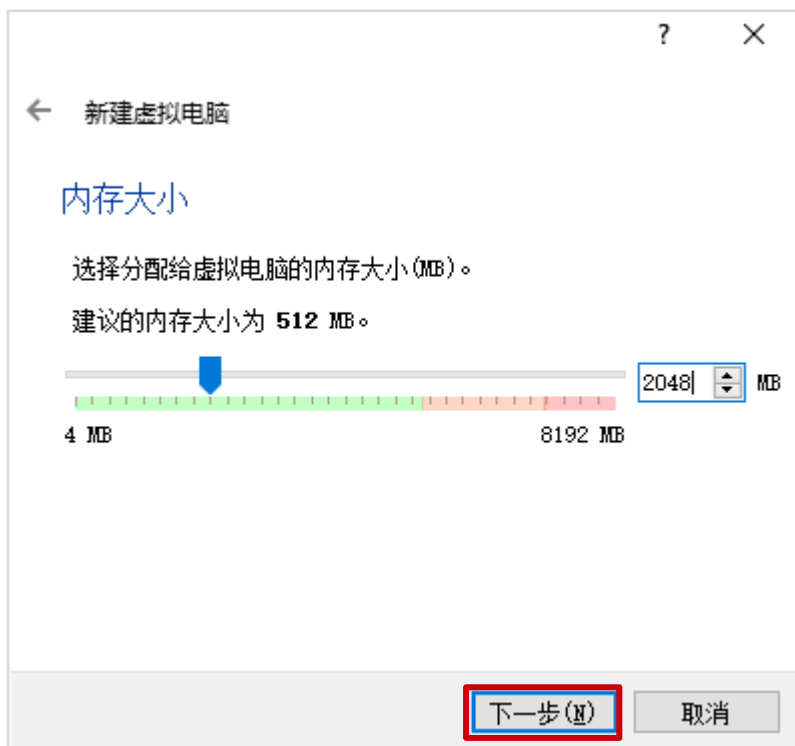
打开 VirtualBox 软件。



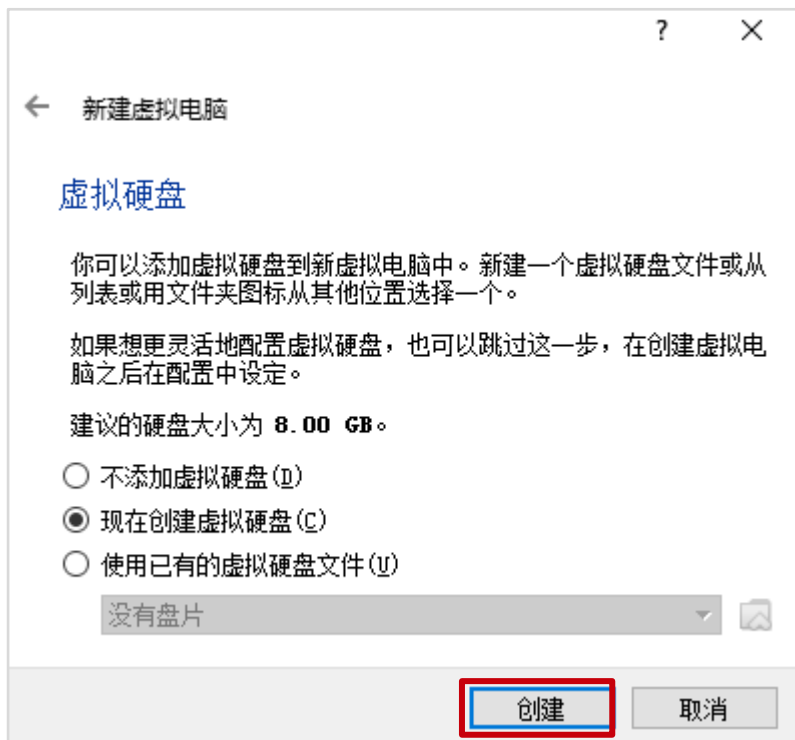
在 VirtualBox 的管理器上点击“新建”，具体如下：



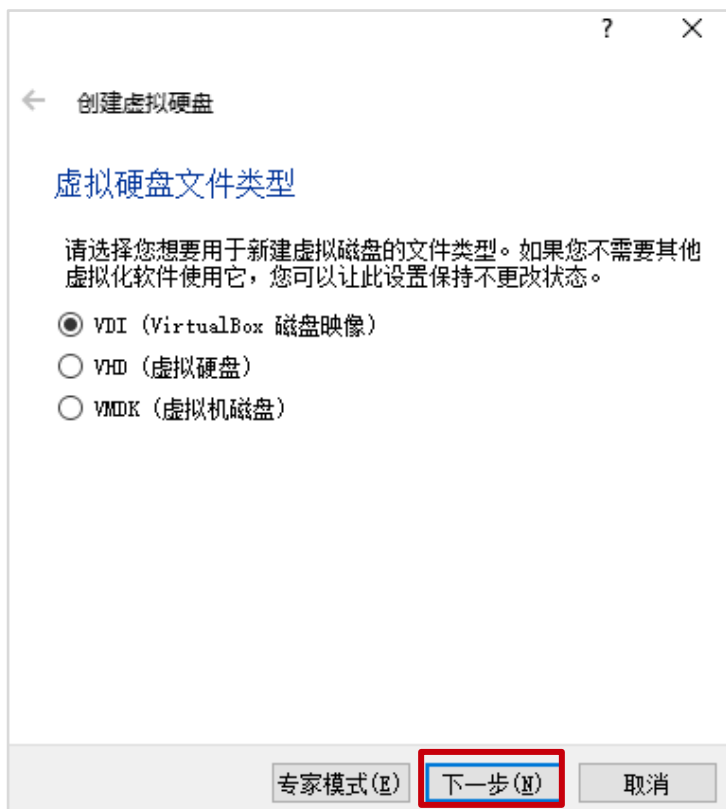
在此页面，填好相应的名称(如：centos)、文件存放地址(如：d:\vmwork)、操作系统类型(如：Linux))、版本(如：Other Linux)，然后点击“下一步”。



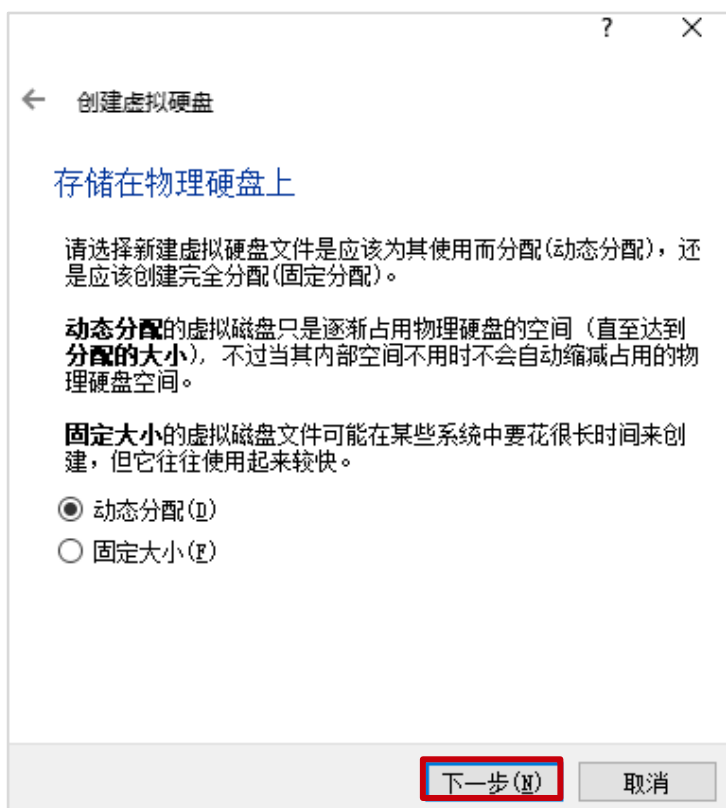
内存设置为 2048MB(表示整个系统只有 2G 内存)，然后点击“下一步”。



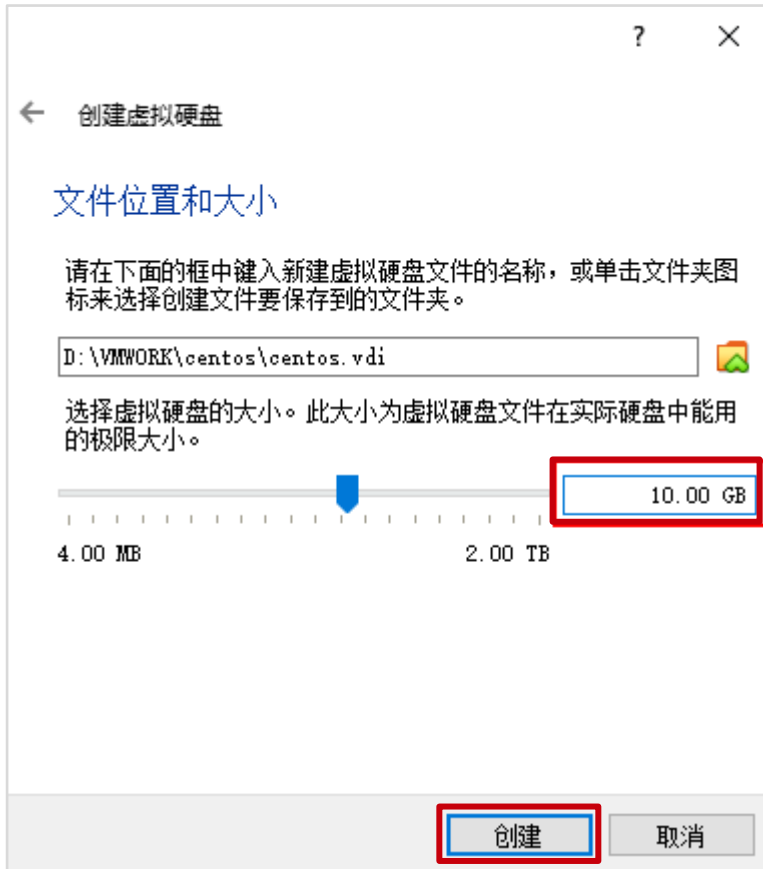
采用默认设置，直接点击“创建”。



采用默认设置，直接点击“下一步”。



采用默认设置，直接点击“下一步”。



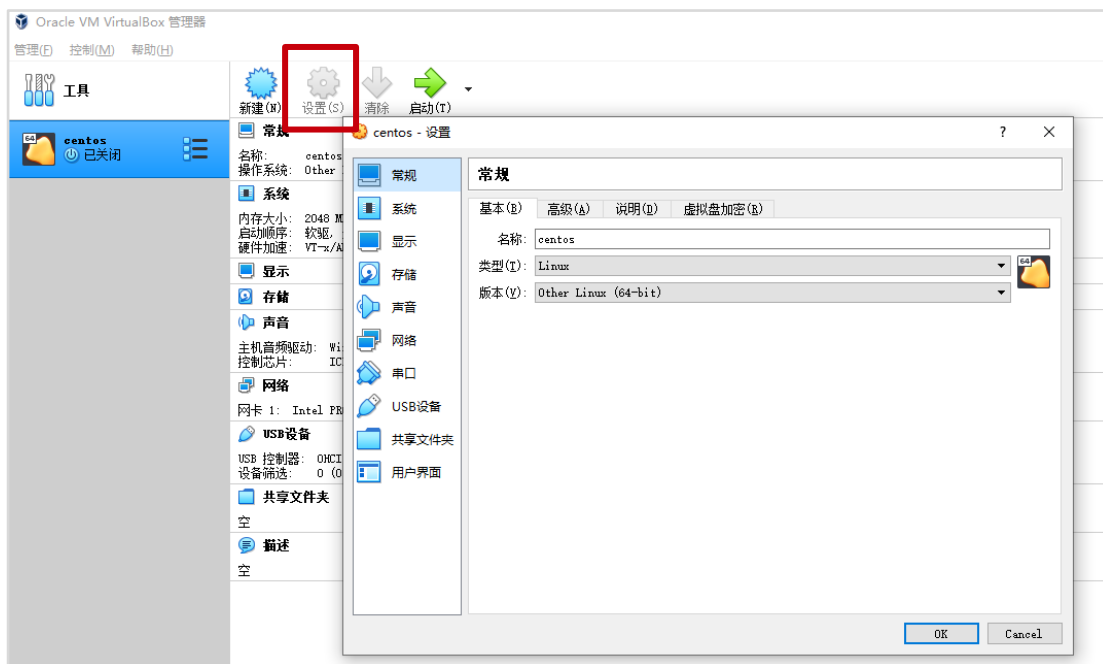
硬盘大小改为 10G（10G 表示硬盘自动扩展后所能达到的最大极限大小，在实际使用中可以根据自己情况来做调整），然后点击“创建”。



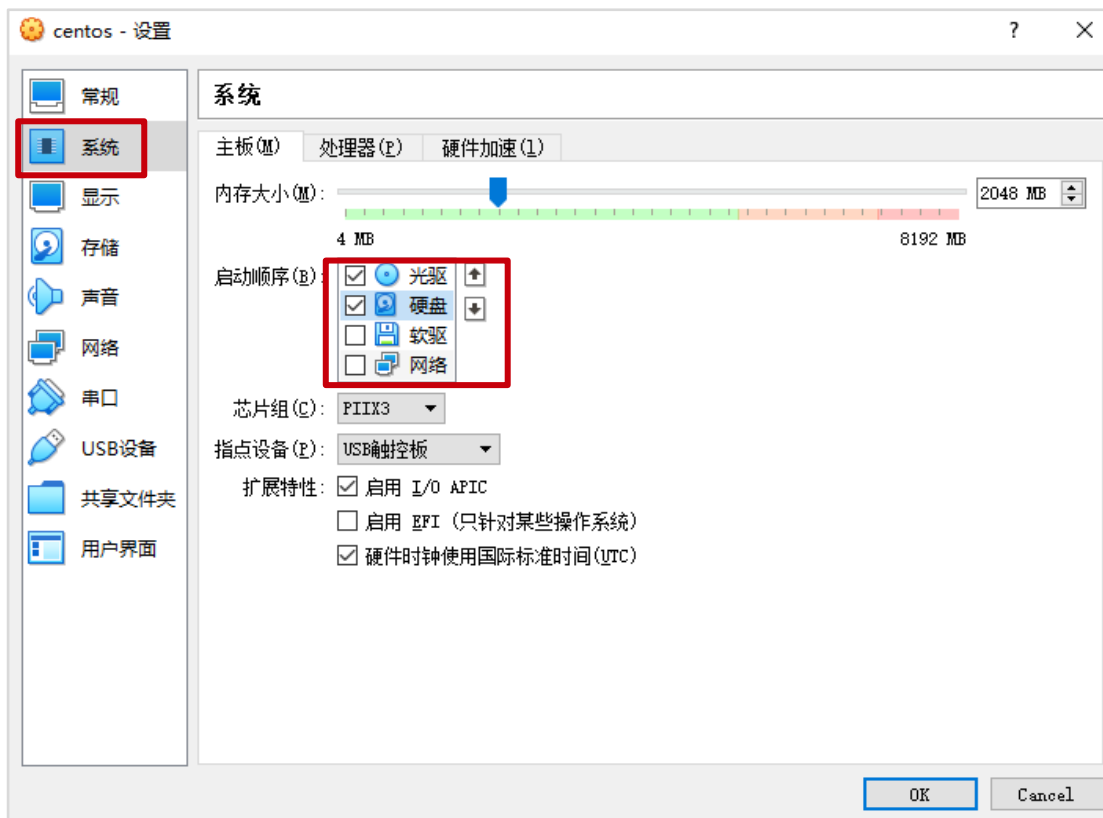
新建虚拟电脑已经完成，接下来对虚拟电脑进行设置。

步骤 2 设置虚拟电脑并安装。

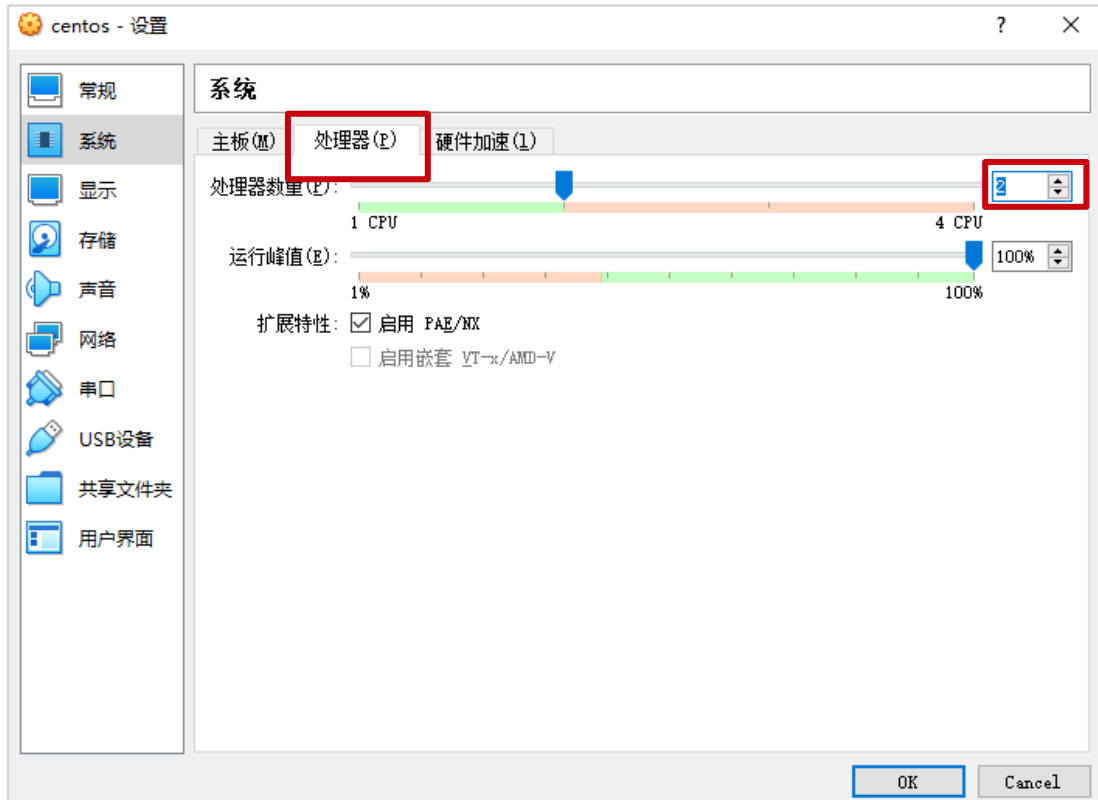
在 VirtualBox 的管理器上点击“设置”，具体如下：



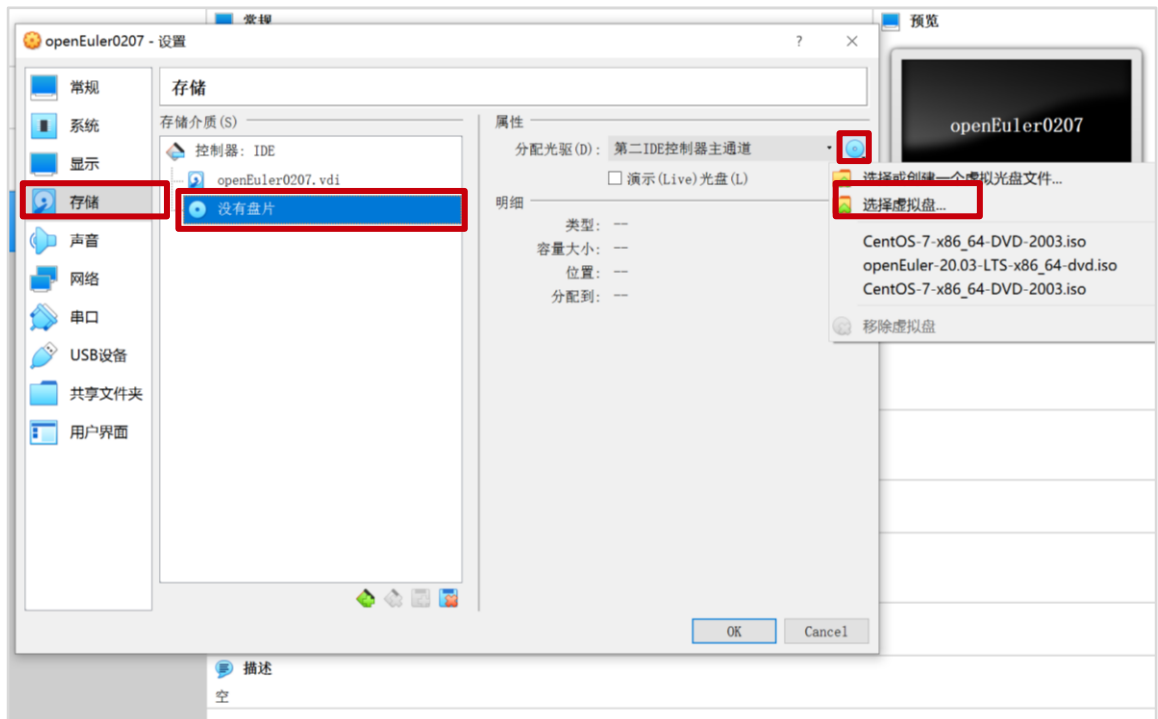
在设置页面上点击“系统”，进行系统设置，在启动顺序中把“光驱”、“硬盘”移动至第一、二启动，“软驱”去掉勾选，具体如下：

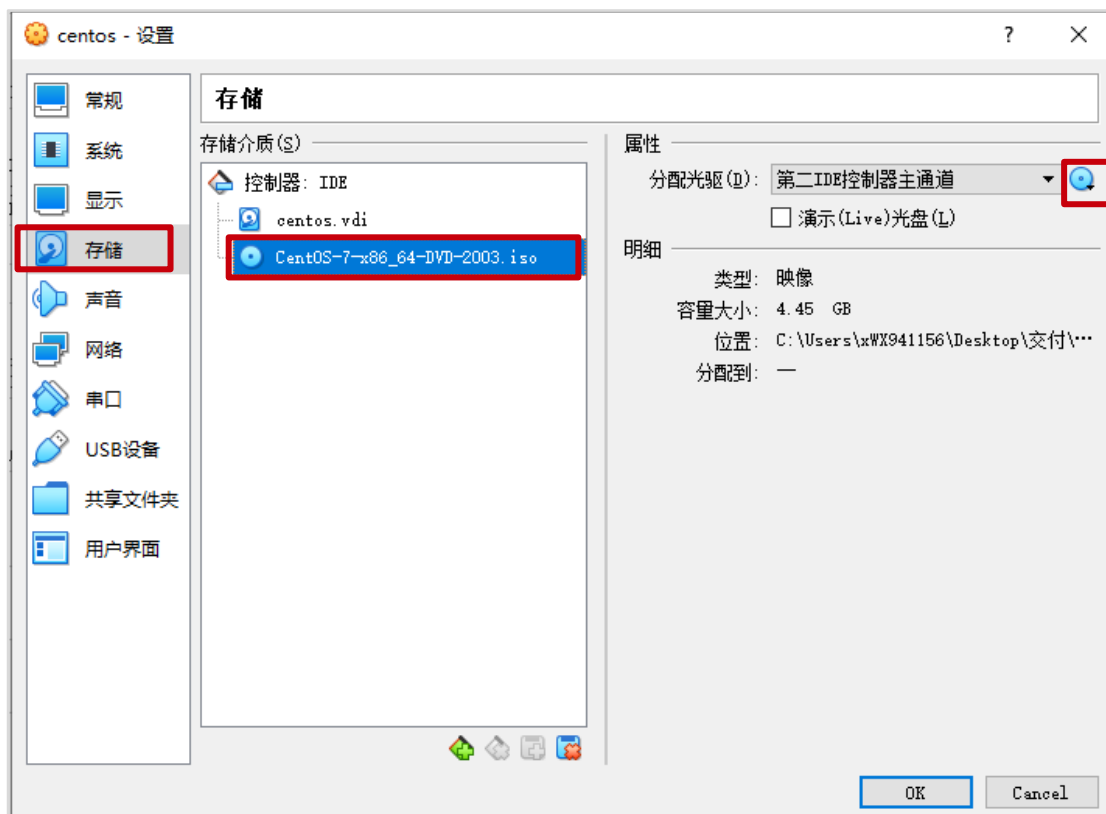


处理器中，CPU 设置为 2，具体如下：

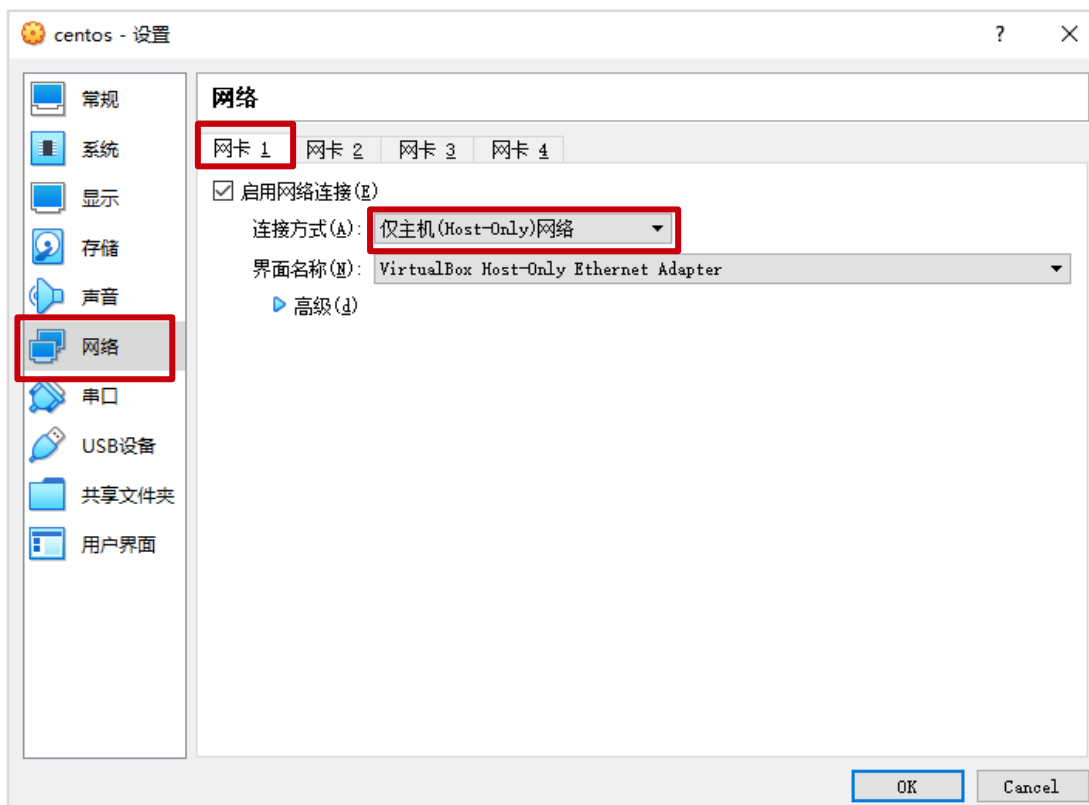


点击“存储”设置项，选择光盘并在分配光驱后面点击加载按钮将前面下载的 centos 7.8 ISO 文件加载至光驱中，具体如下：

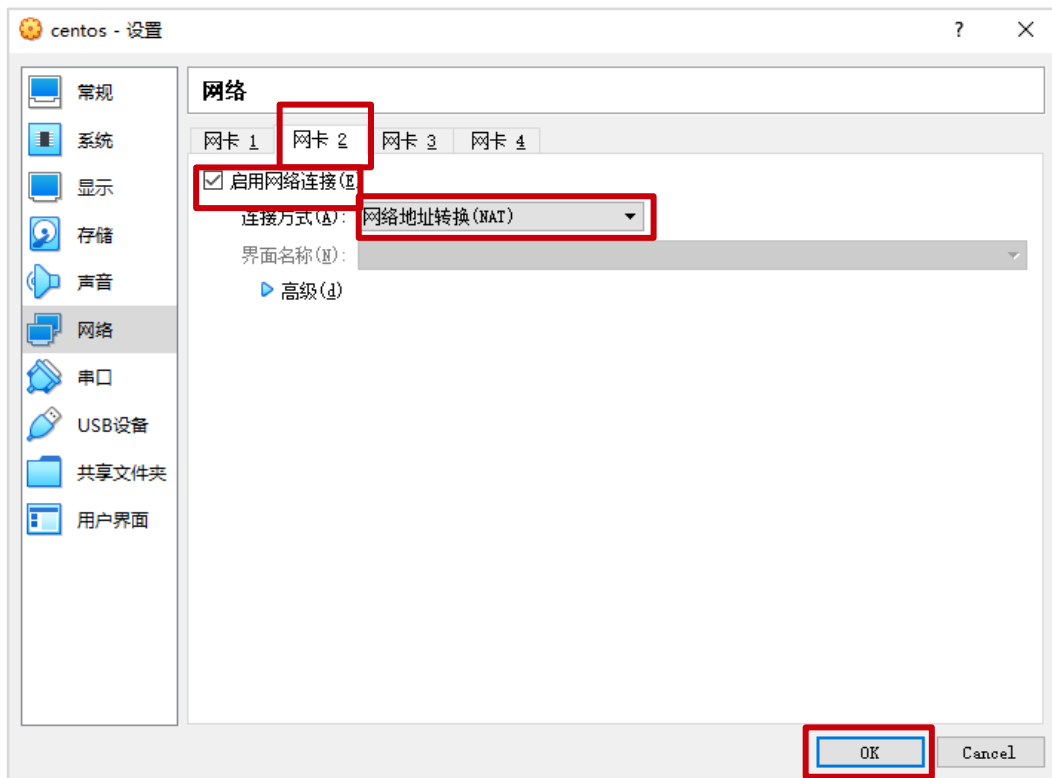




点击“网络”设置项，这里用到二张网卡，网卡 1 连接方式下拉选择为“仅主机 (Host-Only) 网络”，主要用于本地电脑与虚拟电脑相连，具体如下：



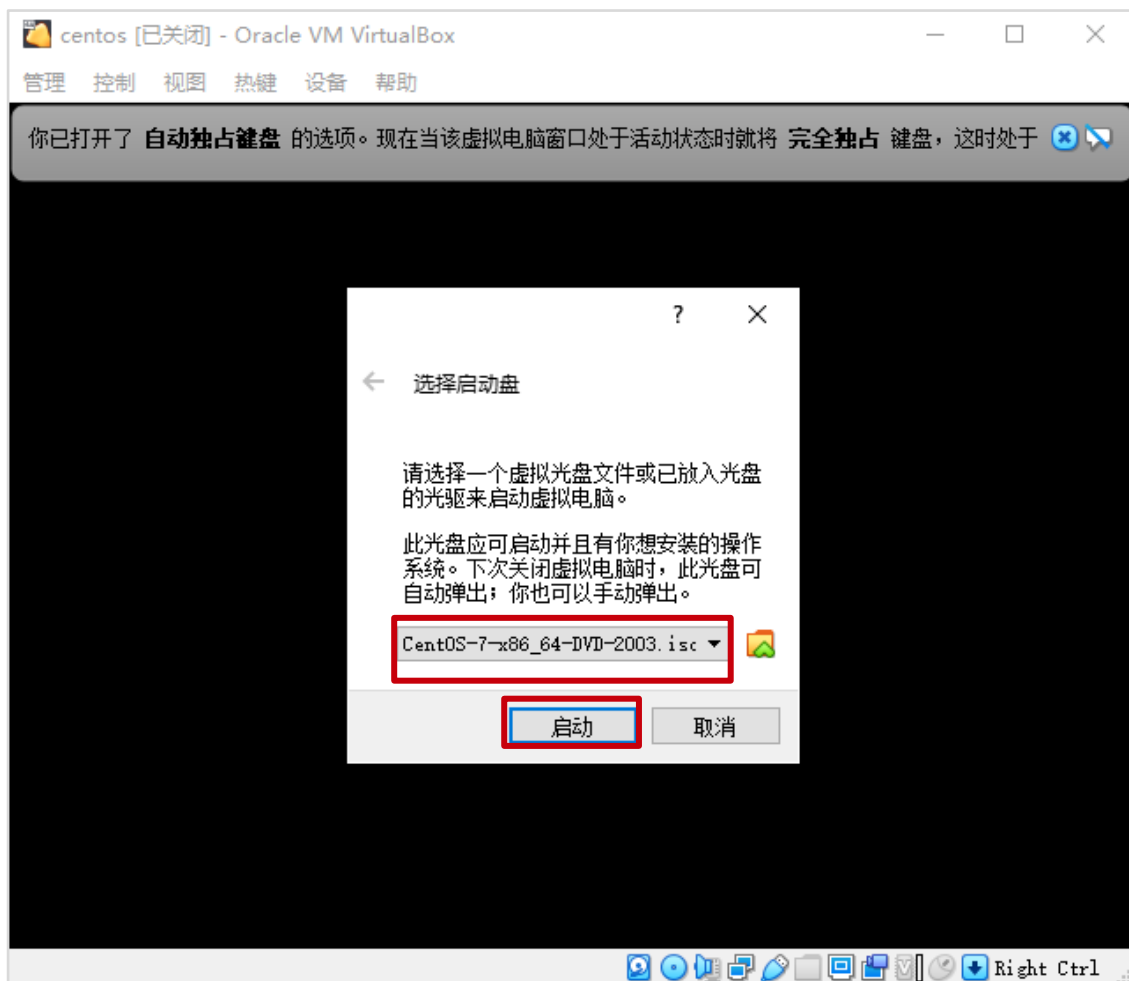
网卡 2 设置，勾选“启用网络连接”，并且“连接方式选择”下拉选择为“网络地址转换(NAT)”，主要用于虚拟电脑与互联网相连，具体如下：



最后点击“OK”按钮，设置完成，设置完成页面如下：



点击启动按钮，开始进入系统安装。

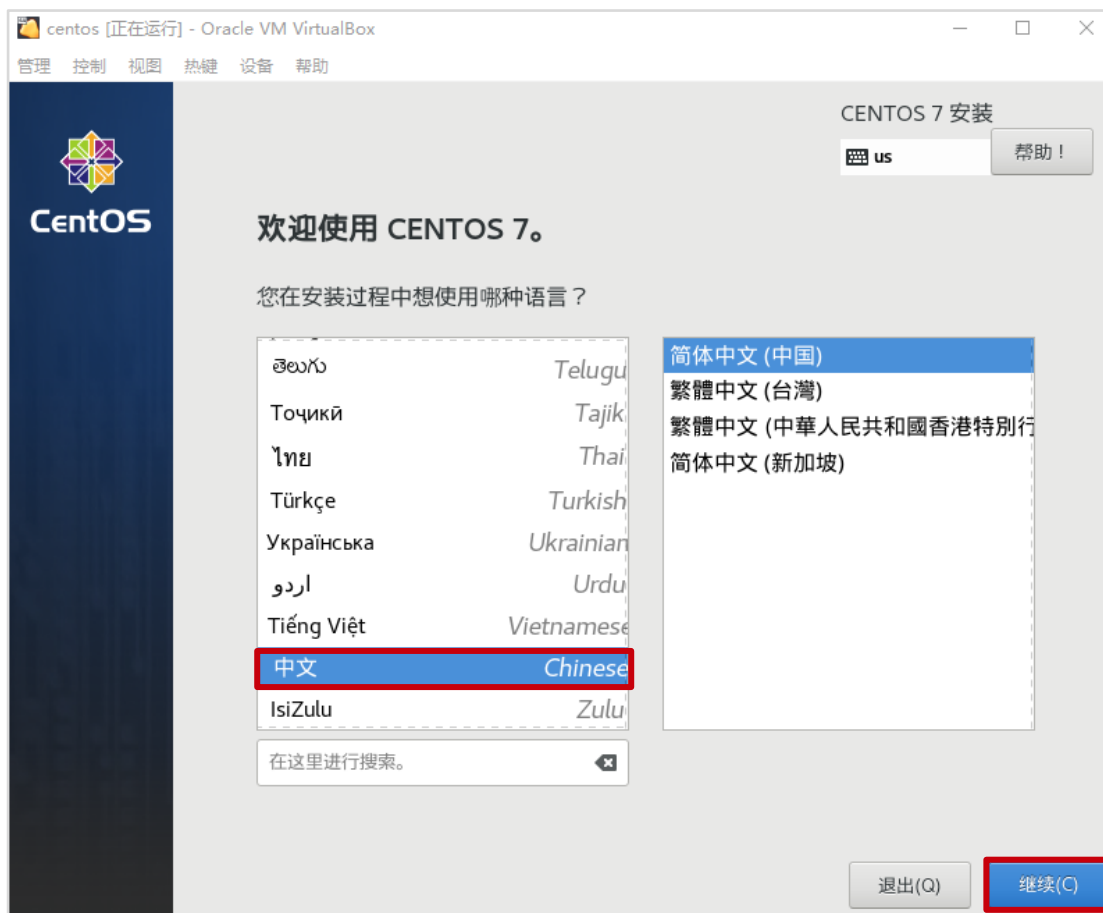


确定光驱中的 ISO 内容正确后，点击“启动”。

启动安装第一界面，直接按下“Enter”键后就会进入自检界面。

```
[ 11.405119] dracut-initqueue[713]: mount: /dev/sr0 is write-protected, mounting read-only
[ OK ] Created slice system-checkisomd5.slice.
Starting Media check on /dev/sr0...
/dev/sr0: 8bb338ea5436a2863746710b44ac3540
Fragment sums: 5d2292bff63cc47ee24ffbe453c08fb9276e3fda752d628b44ec514dfebe
Fragment count: 20
Press [Esc] to abort check.
Checking: 019.9%
```

在自检界面按下“Esc”键跳过自检，然后进入如下界面：



语言选择中文----》简体中文（中国），然后点击“继续”。



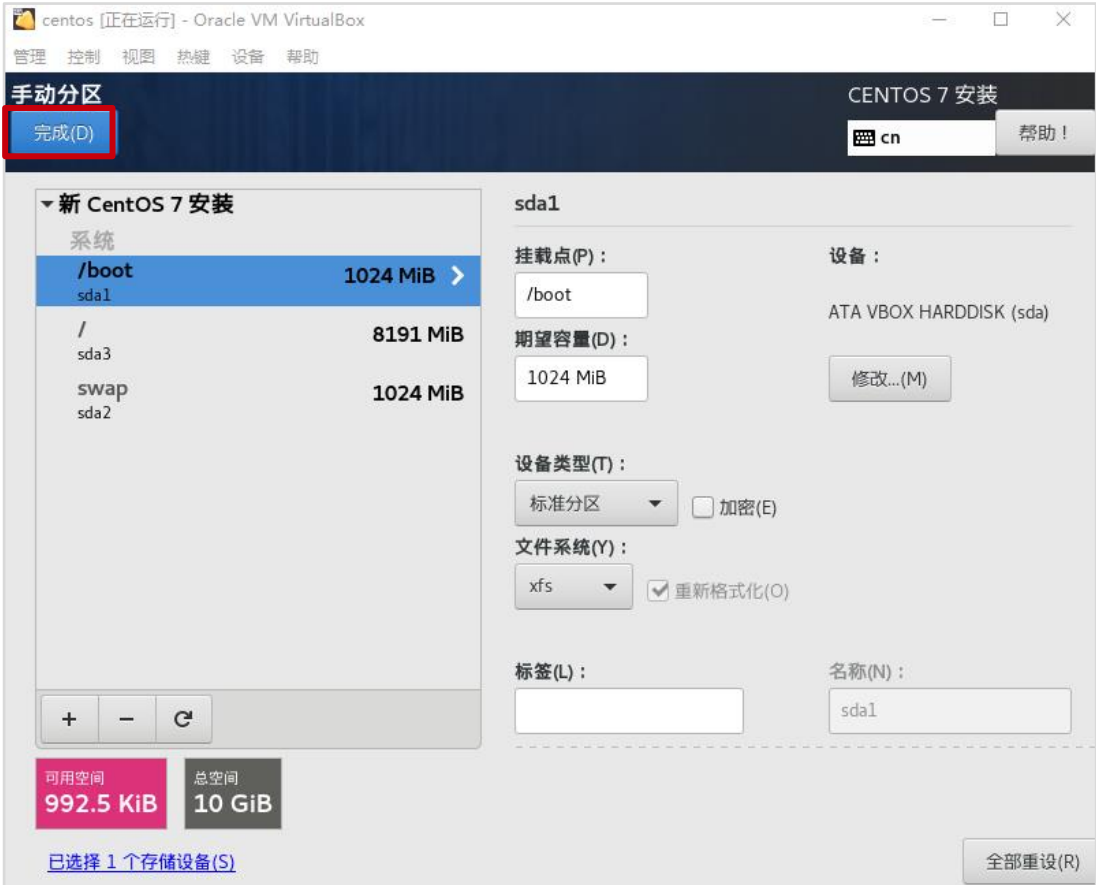
在此页面，点击“安装位置”先进行分区，具体如下：



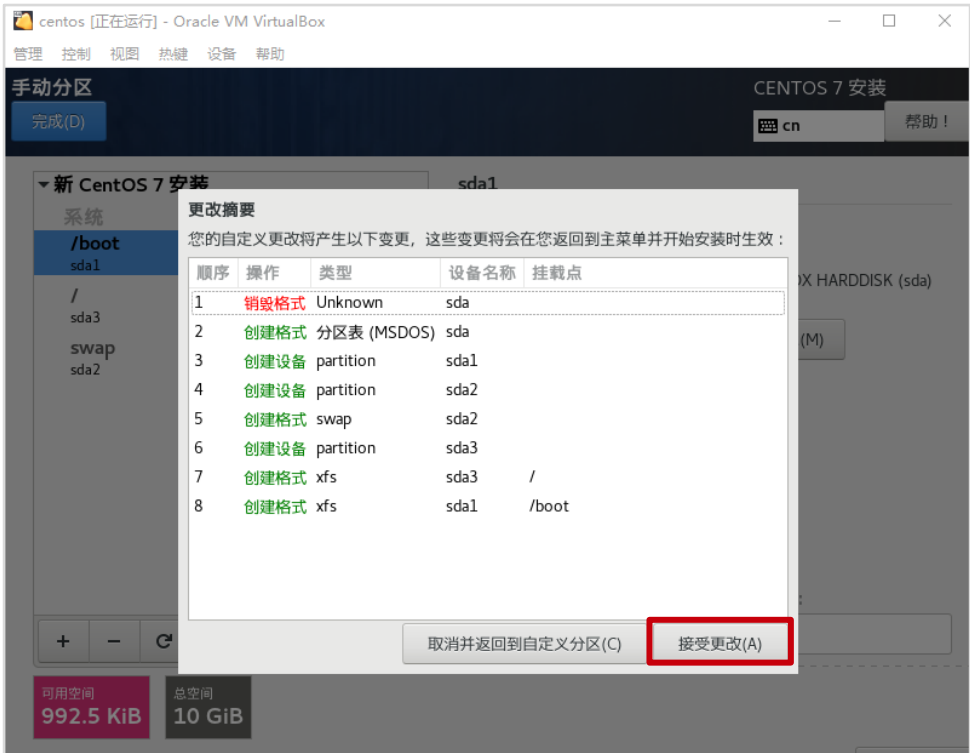
选择“我要配置分区”，然后点击“完成”。



下拉选择“标准分区”，然后点击“点这里自动创建他们”。

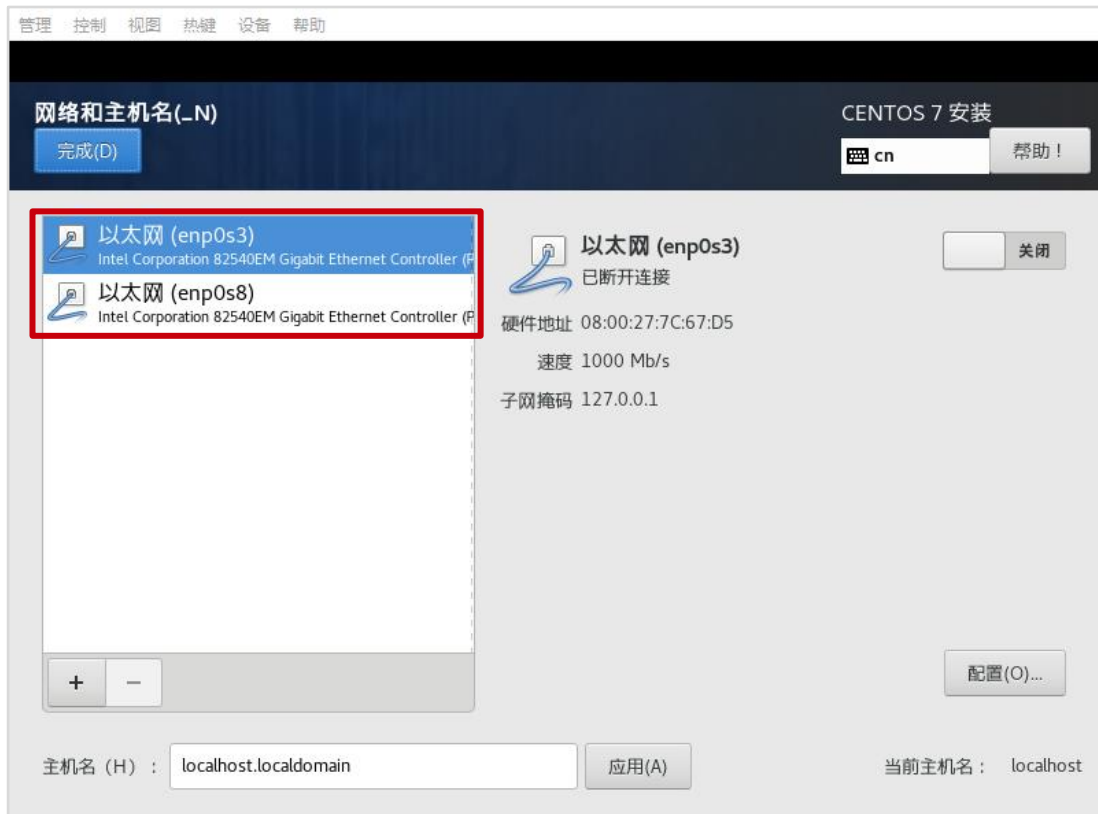


点击“完成”，进入下一步设置。



点击“接受更改”，完成分区设置，接下来进行网络和主机名设置。

在安装信息摘要页面，点击“网络和主机名”进行网络和主机名设置，具体如下：

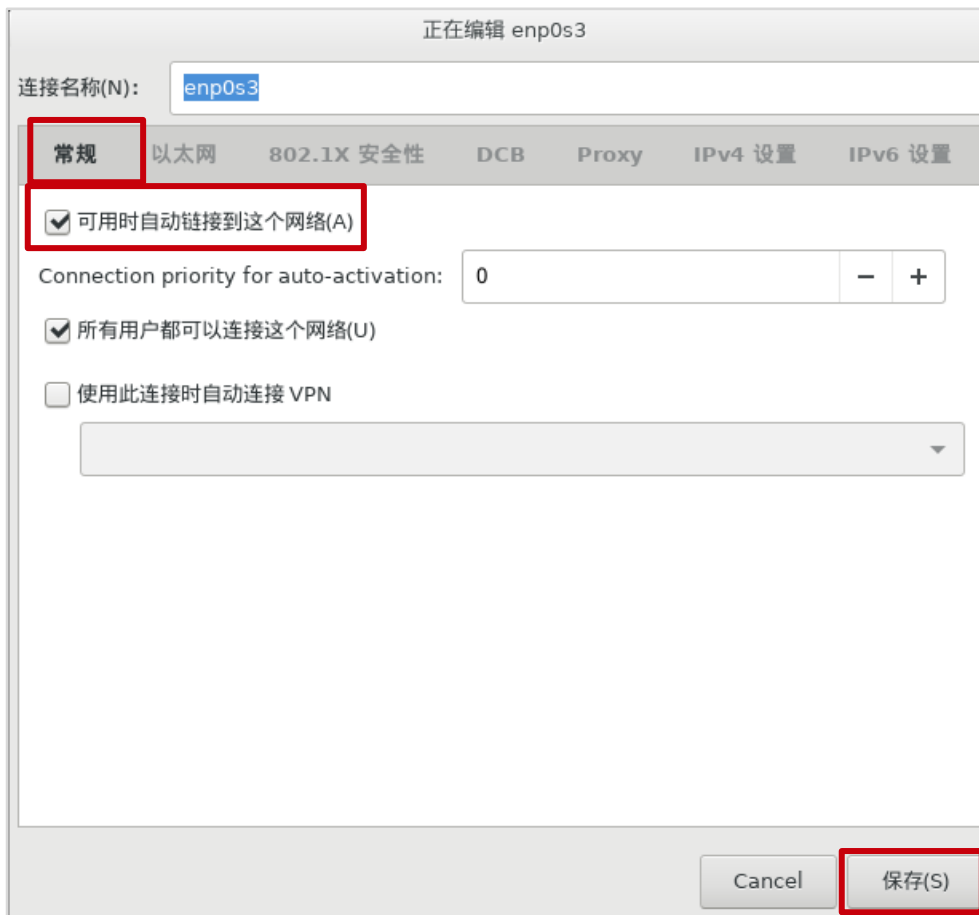


分别针对二张网卡进行设置，如以太网 (enp0s3) 网卡，先点击“关闭”边上的按钮把网卡打开。

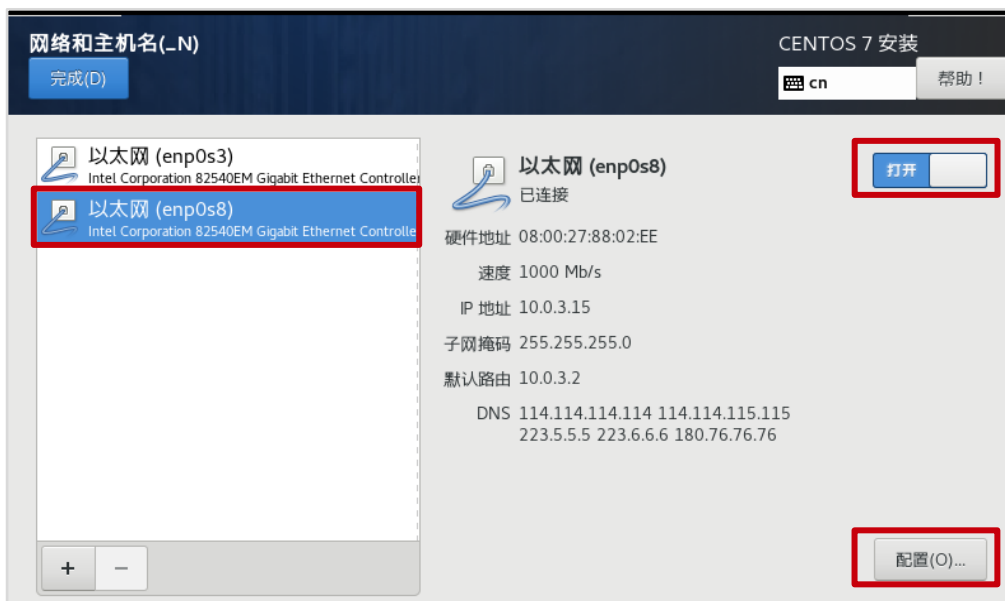


接着设置主机名（如：db1），并点击“应用（A）”，然后点击“配置”。

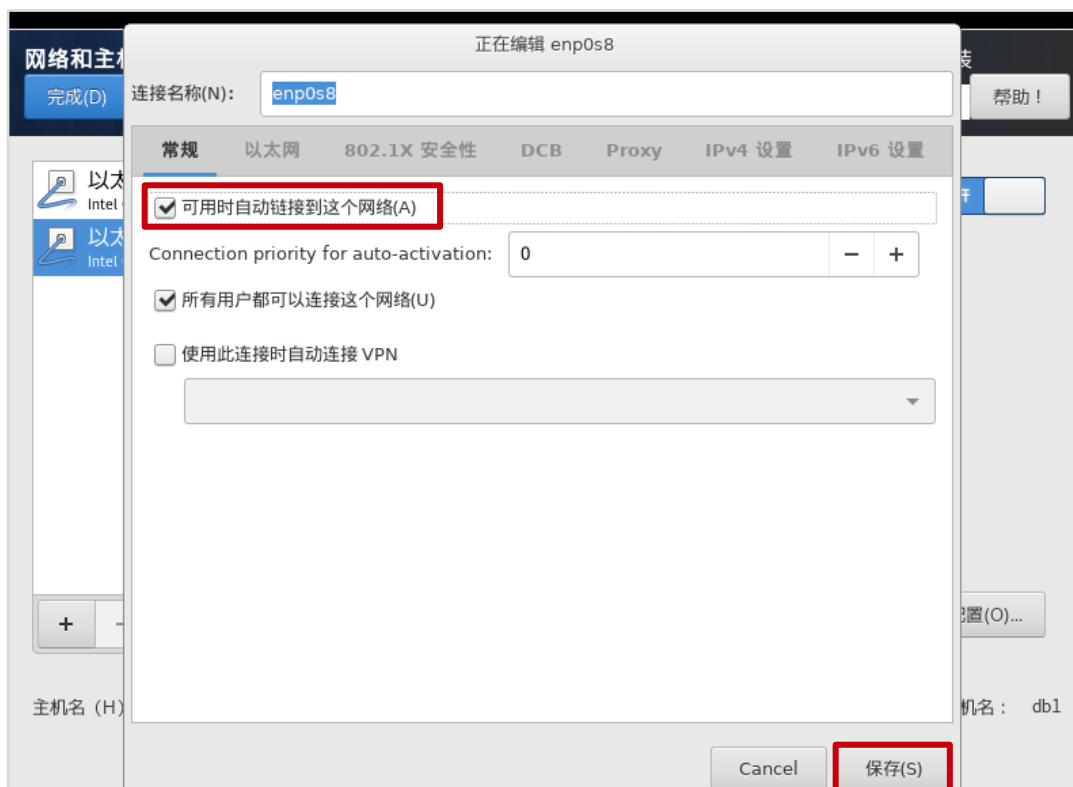
说明：设置主机名时一定要注意，如果在同一网段内有多位学员按此文档来安装，请尽量把主机名设成不一样。



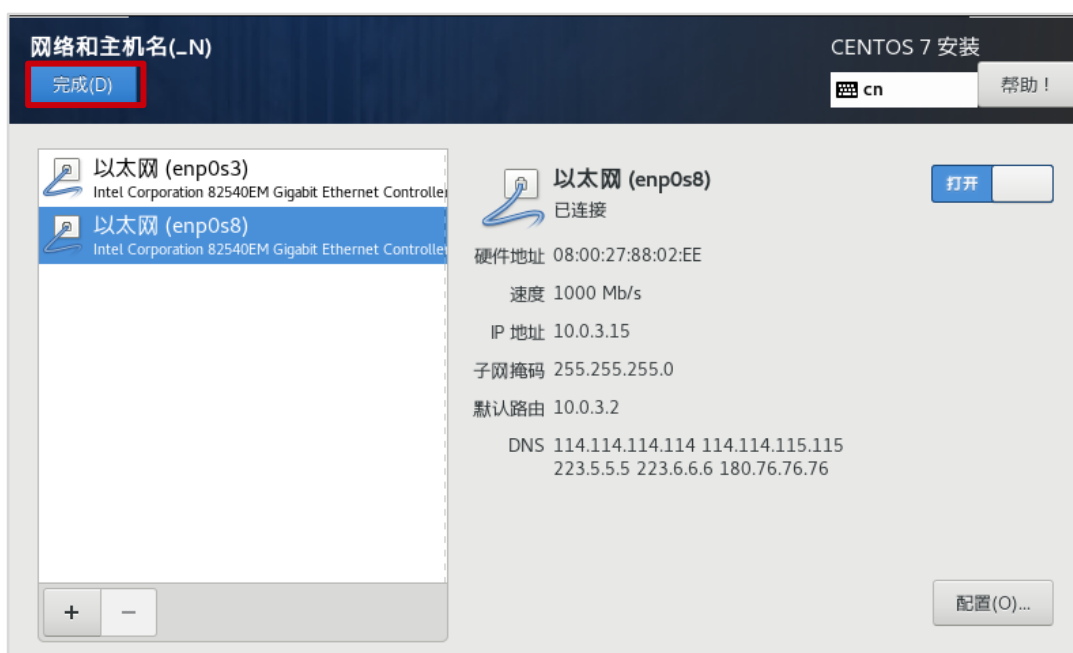
在配置页中，选择“常规”，然后勾选“可用时自动链接到这个网络”，接着点击“保存”。



参数第一张网卡的设置步骤对第二张网卡进行设置，先打开网卡，接着进行配置，具体如下：



点击“保存”。



点击“完成”，结束网络和主机名的设置。

在安装信息摘要页面，点击“软件选择”进行软件安装设置，具体如下：



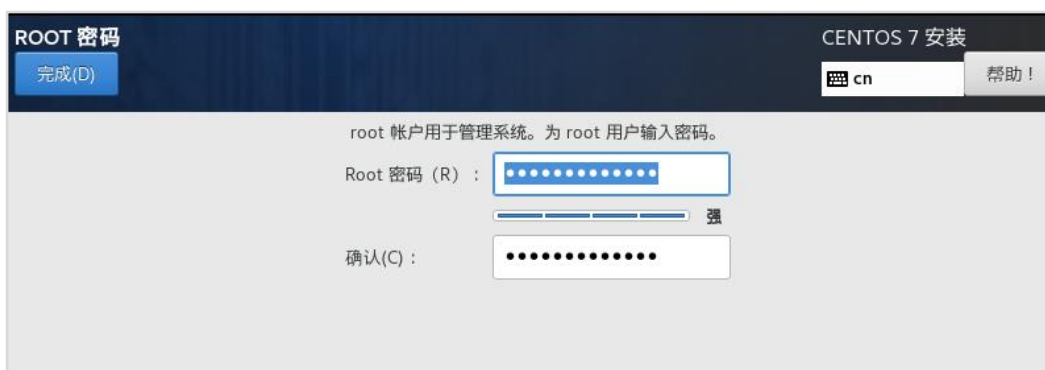
在此页面选择“虚拟化主机”，并在右边勾选“Linux 的远程管理”、“开发工具”、“安全性工具”、“系统管理工具”。然后点击完成。



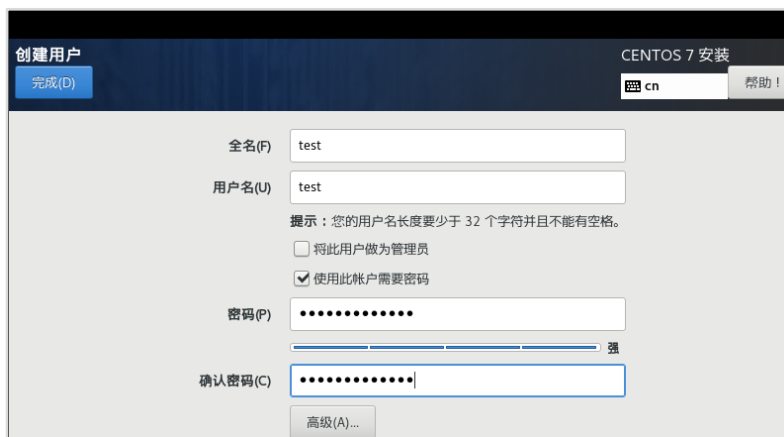
设置结束，点击“开始安装”，具体如下：



点击“ROOT 密码”，给 ROOT 用户设置密码（如：openGauss@123）。

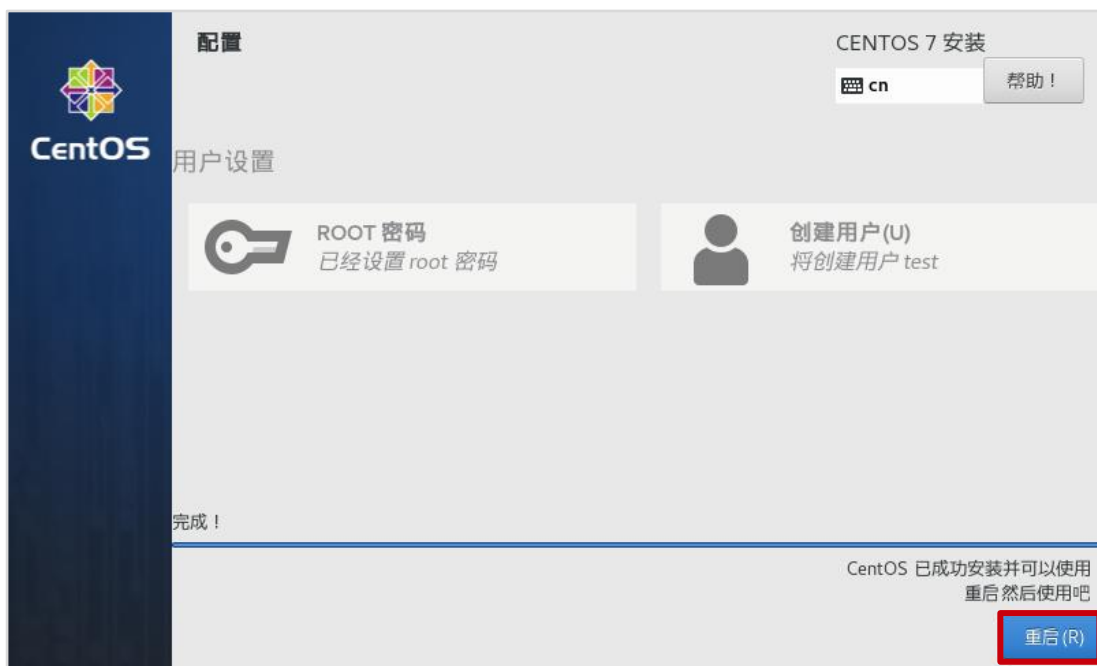


点击“创建用户”，在此新创建一个用户（如：用户 test，密码 openGauss@123），具体如下：

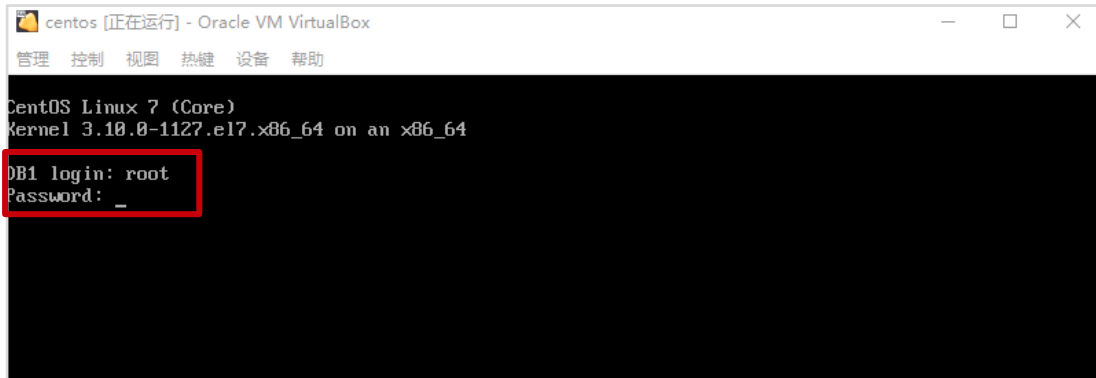




点击“完成配置”，系统安装中，等待数分钟后会出现如下界面：



出现此界面表示，系统安装完成，然后点击“重启”。



重启后，要求输入用户名及密码，这里用 root 用户登录，密码比如：openGauss@123。

注意：命令行环境下，输入密码没有回显，输入的内容不显示在界面上，直接输入密码后回车即可。

步骤 3 确认网络。

在 Linux 操作系统上，通过 ifconfig 来查看二张网卡是否都正常启动，具体如下：

```
[root@db1 ~]# ifconfig
enpos3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.108 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::ac2f:dc4f:edfe:1d57 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:0f:78:e3 txqueuelen 1000 (Ethernet)
    RX packets 519 bytes 48509 (47.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 178 bytes 52937 (51.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enpos8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.3.15 netmask 255.255.255.0 broadcast 10.0.3.255
    inet6 fe80::bedc:2040:4b9:23ed prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:45:8d:fo txqueuelen 1000 (Ethernet)
    RX packets 72 bytes 10702 (10.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 124 bytes 11664 (11.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

.....
virbro: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 192.168.122.1 netmask 255.255.255.0 broadcast 192.168.122.255
    ether 52:54:00:05:11:90 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

通过 ping baidu.com 确认是否能上网，具体如下：

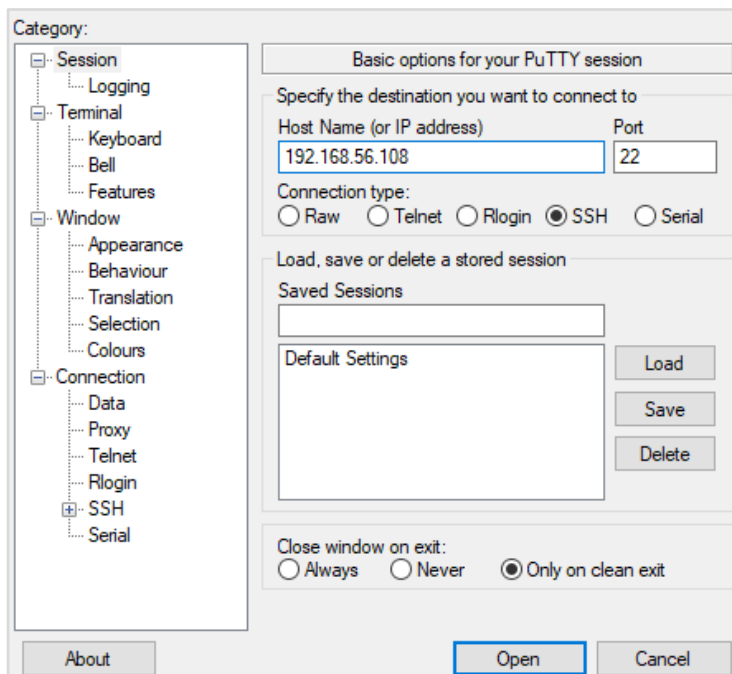
```
[root@db1 ~]# ping baidu.com
PING baidu.com (39.156.69.79) 56(84) bytes of data.
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=1 ttl=47 time=48.1 ms
```

```
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=2 ttl=47 time=46.5 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=3 ttl=47 time=49.2 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=4 ttl=47 time=47.3 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=5 ttl=47 time=46.7 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=6 ttl=47 time=45.9 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=7 ttl=47 time=46.7 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=8 ttl=47 time=48.3 ms
```

使用 `ctrl+c` 组合键可以结束输出。

1.5 操作系统环境准备

为了操作方便，可以使用 SSH 工具（比如：PuTTY 等）从本地电脑通过配置 `enpos3` 网卡的 IP 地址（如：`192.168.56.108`）来连接虚拟机，并使用 ROOT 用户来登录。



步骤 1 修改操作系统版本。

如果 CentOS 版本不是 7.6 的需要进行修改，如果是 7.6 则无需修改，

先 `vi /etc/redhat-release` 打开编辑文件，然后将内容改为 CentOS Linux release 7.6.2003 (Core)。输入“i”切换到编辑模式，移动鼠标到修改位置修改内容，然后按下 ESC 键退出编辑模式，然后输入“:wq”退出并进行保存，具体如下：

```
[root@db1 ~]# vi /etc/redhat-release

CentOS Linux release 7.6.2003 (Core)
```

步骤 2 关闭防火墙。

执行以下二个命令将防火墙关闭，

systemctl stop firewalld.service

systemctl disable firewalld.service，具体如下：

```
[root@db1 ~]# systemctl stop firewalld.service
[root@db1 ~]# systemctl disable firewalld.service
Removed symlink /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
[root@db1 ~]#
```

步骤 3 设置字符集及环境变量。

```
[root@db1 ~]# cat >>/etc/profile<<EOF
export LANG=en_US.UTF - 8
EOF
[root@db1 ~]#
[root@db1 ~]# cat >>/etc/profile<<EOF
export packagePath=/opt/software/openGauss
EOF
[root@db1 ~]# source /etc/profile
[root@db1 ~]# cat >>/etc/profile<<EOF
export LD_LIBRARY_PATH=$packagePath/script/gspylib/clib:$LD_LIBRARY_PATH
EOF
[root@db1 ~]#
[root@db1 ~]# source /etc/profile
```

验证变量是否生效。

```
[root@db1 ~]# echo $LD_LIBRARY_PATH

/opt/software/openGauss/script/gspylib/clib:
```

步骤 4 关闭 swap 交换内存。

```
[root@db1 ~]# swapoff -a
```

步骤 5 准备 yum 环境。

备份原有的 yum 配置文件。

```
[root@db1 ~]# mv /etc/yum.repos.d/CentOS-Base.repo /etc/yum.repos.d/CentOS-Base.repo.bak
[root@db1 ~]#
```

下载可用源的 repo 文件,可通过以下二种方式下载：

方式一：

```
curl -o /etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-7.repo
```

方式二：

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

```
[root@db1 ~]# curl -o /etc/yum.repos.d/CentOS-Base.repo https://mirrors.huaweicloud.com/repository/conf/CentOS-7-anon.repo
  %Total    %Received %Xferd   Average Speed   Time    Time       Time  Current
             Dload  Upload  Total    Spent    Left  Speed
100  2523  100  2523    0     0  14065      0 --:--:-- --:--:-- --:--:-- 14094
```

查看 repo 文件内容是否正确，如果显示的内容不正确，请选择另一种方式下载可用源的 repo 文件。

```
[root@db1 ~]# cat /etc/yum.repos.d/CentOS-Base.repo
# CentOS-Base.repo
#
# The mirror system uses the connecting IP address of the client and the
# update status of each mirror to pick mirrors that are updated to and
# geographically close to the client.  You should use this for CentOS updates
# unless you are manually picking other mirrors.
#
# If the mirrorlist= does not work for you, as a fall back you can try the
# remarked out baseurl= line instead.
#
[base]
name=CentOS-$releasever - Base - mirrors.aliyun.com
failovermethod=priority
baseurl=http://mirrors.aliyun.com/centos/$releasever/os/$basearch/
        http://mirrors.aliyuncs.com/centos/$releasever/os/$basearch/
        http://mirrors.cloud.aliyuncs.com/centos/$releasever/os/$basearch/
gpgcheck=1
.....
```

步骤 6 yum 安装相关包。

执行以下命令，安装所需的包

```
yum install -y libaio-devel flex bison ncurses-devel glibc-devel patch lsb_release wget python3
```

如下：

```
[root@db1 ~]# yum install -y libaio-devel flex bison ncurses-devel glibc-devel patch lsb_release wget python3
Loaded plugins: fastestmirror, langpacks
Determining fastest mirrors
base                | 3.6 kB  00:00:00
extras              | 2.9 kB  00:00:00
updates             | 2.9 kB  00:00:00
.....
```

步骤 7 设置默认 Python 版本为 3.x。

```
[root@db1 ~]# cd /usr/bin
[root@db1 bin]# mv python python.bak
[root@db1 bin]# ln -s python3 /usr/bin/python
[root@db1 bin]# python -V
Python 3.6.8
```

修改完成后，需要确认 yum 是否能使用，如果不能使用需要修改/usr/bin/yum 文件，把 #!/usr/bin/python 这行修改为 #!/usr/bin/python2.7（或者对应的 python 2.x 的版本）。输入“i”切换到编辑模式，移动鼠标到修改位置修改内容，然后按下 ESC 键退出编辑模式，然后输入“:wq”退出并进行保存。如下：

```
[root@db1 bin]# vi /usr/bin/yum
```

```
#!/usr/bin/python2.7
import sys
.....
```

用 yum --help 命令来验证 yum 是否能使用。

```
[root@db1 bin]# yum --help
```

```
Loaded plugins: fastestmirror, langpacks
```

```
Usage: yum [options] COMMAND
```

```
List of Commands:
```

```
check          Check for problems in the rpmdb
```

```
check-update    Check for available package updates
.....
```

步骤 8 创建存放数据库安装目录。

```
[root@db1 ~]# mkdir -p /opt/software/openGauss
```

```
[root@db1 ~]# cd /opt/software/openGauss
```

```
[root@db1 openGauss]#
```

步骤 9 下载数据库安装包。

使用 wget 下载数据库安装包，

```
cd /root
```

```
wget https://opengauss.obs.cn-south-1.myhuaweicloud.com/1.1.0/x86/openGauss-1.1.0-CentOS-64bit-all.tar.gz
```

具体如下：

```
[root@db1 openGauss]# wget https://opengauss.obs.cn-south-1.myhuaweicloud.com/1.1.0/x86/openGauss-1.1.0-CentOS-64bit-all.tar.gz
```

```
--2020-09-10 14:23:07-- https://opengauss.obs.cn-south-1.myhuaweicloud.com/1.1.0/x86/openGauss-1.1.0-CentOS-64bit-all.tar.gz
```

1.6 安装 openGauss 数据库

步骤 1 创建 XML 配置文件，用于数据库安装。

```
[root@db1 openGauss]# vi clusterconfig.xml
```

将以下内容添加进 clusterconfig.xml 文件中。输入“i”切换到编辑模式，复制内容黏贴到文档中，然后按下 ESC 键退出编辑模式，然后输入“:wq”退出并进行保存。

```
<?xml version="1.0" encoding="UTF-8"?>
<ROOT>
  <!-- openGauss 整体信息 -->
  <CLUSTER>
    <PARAM name="clusterName" value="dbCluster" />
    <PARAM name="nodeNames" value="db1" />
    <PARAM name="backIp1s" value="10.0.3.15"/>
    <PARAM name="gaussdbAppPath" value="/opt/gaussdb/app" />
    <PARAM name="gaussdbLogPath" value="/var/log/gaussdb" />
    <PARAM name="gaussdbToolPath" value="/opt/huawei/wisquery" />
    <PARAM name="corePath" value="/opt/opengauss/corefile"/>
    <PARAM name="clusterType" value="single-inst"/>
  </CLUSTER>
  <!-- 每台服务器上的节点部署信息 -->
  <DEVICELIST>
    <!-- node1 上的节点部署信息 -->
    <DEVICE sn="1000001">
      <PARAM name="name" value="db1"/>
      <PARAM name="azName" value="AZ1"/>
      <PARAM name="azPriority" value="1"/>
      <!-- 如果服务器只有一个网卡可用，将 backIp1 和 sshIp1 配置成同一个 IP -->
      <PARAM name="backIp1" value="10.0.3.15"/>
      <PARAM name="sshIp1" value="10.0.3.15"/>

      <!--dbnode-->
      <PARAM name="dataNum" value="1"/>
      <PARAM name="dataPortBase" value="26000"/>
      <PARAM name="dataNode1" value="/gaussdb/data/db1"/>
    </DEVICE>
  </DEVICELIST>
</ROOT>
```

说明：其中标红的内容，需要根据自己实际的 IP 和主机名进行修改，如果其中的中文出现乱码时可以删除这些行。

步骤 2 将下载好的安装包解压至存放目录。

先解压 openGauss-1.1.0-CentOS-64bit-all.tar.gz 包。

```
[root@db1 openGauss]# tar -zxvf openGauss-1.1.0-CentOS-64bit-all.tar.gz
```

再先解压 openGauss-1.1.0-CentOS-64bit-om.tar.gz 包。

```
[root@db1 openGauss]# tar -zxvf openGauss-1.1.0-CentOS-64bit-om.tar.gz
```

解压后如下，用 ls 命令查看如下：

```
[root@db1 openGauss]# ls
clusterconfig.xml      openGauss-Package-bak_392c0438.tar.gz
```

lib	script
openGauss-1.1.0-CentOS-64bit-all.tar.gz	simpleInstall
openGauss-1.1.0-CentOS-64bit-om.sha256	upgrade_sql.sha256
openGauss-1.1.0-CentOS-64bit-om.tar.gz	upgrade_sql.tar.gz
openGauss-1.1.0-CentOS-64bit.sha256	version.cfg
openGauss-1.1.0-CentOS-64bit.tar.bz2	

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。

更改权限。

```
[root@db1 openGauss]# chmod 755 -R /opt/software
[root@db1 openGauss]#
```

步骤 3 执行初始化脚本。

```
[root@db1 openGauss]# cd /opt/software/openGauss/script
[root@db1 script]# python gs_preinstall -U omm -G dbgrp -X /opt/software/openGauss/clusterconfig.xml
Parsing the configuration file.
Successfully parsed the configuration file.
Installing the tools on the local node.
Are you sure you want to create trust for root (yes/no)? yes
Please enter password for root.
Password:
Creating SSH trust for the root permission user.
Checking network information.
.....
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:
Please enter password for cluster user again.
Password:
Successfully created [omm] user on all nodes.
Preparing SSH service.
Successfully prepared SSH service.
.....
Successfully set finish flag.
Preinstallation succeeded.
```

期间需要输入操作系统 root 用户的密码（如密码：openGauss@123）和创建操作系统 omm 用户及设置密码（如密码：openGauss@123）。密码依然不回显，直接输入密码并回车即可。

当返回 Preinstallation succeeded 内容时，表明初始化完成。

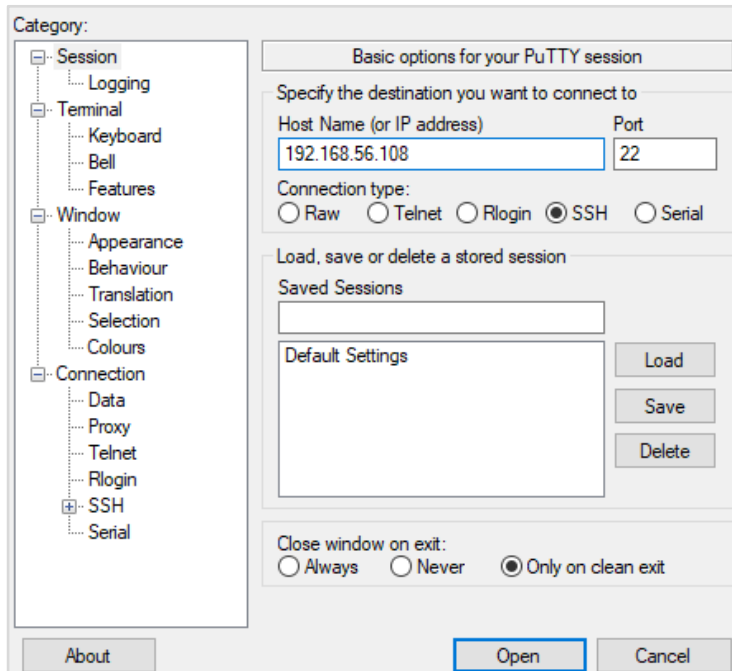
步骤 4 初始化数据库。

用 init 6 重启下虚拟机（主要是为了释放一些内存资源）。

```
[root@db1 script]# init 6
Connection closing...Socket close.
Connection closed by foreign host.
Disconnected from remote host(ONE) at 10:51:59.
```

Type 'help' to learn how to use Xshell prompt.

过段时间虚拟机重启好后，再次使用 SSH 工具（比如：PuTTY 等）从本地电脑通过配置 enpos3 网卡的 IP 地址（如：**192.168.56.108**）来连接虚拟机，并使用 ROOT 用户来登录，然后接着以下操作。



先更新下权限。

```
[root@db1 script]# chmod 755 -R /opt/software
[root@db1 openGauss]#
```

然后使用 omm 用户进行数据库初始化。

注意：根据用户实际内存大小设置对应的共享内存的大小，如果对该参数进行了设置，会在数据库启动时候报错，本实验虚拟机总内存大小是 2G。

```
gs_install -X /opt/software/openGauss/clusterconfig.xml --gsinit-parameter="--encoding=UTF8" --
dn-guc="max_process_memory=2GB" --dn-guc="shared_buffers=128MB" --dn-
guc="bulk_write_ring_size=128MB" --dn-guc="cstore_buffers=16MB"
```

具体如下：

```
[root@db1 openGauss]# su - omm
Last login: Thu Sep 10 15:26:21 CST 2020 on pts/o
[omm@db1 ~]$ cd /opt/software/openGauss/script
[omm@db1 script]$ gs_install -X /opt/software/openGauss/clusterconfig.xml --gsinit-parameter="--encoding=UTF8"
--dn-guc="max_process_memory=2GB" --dn-guc="shared_buffers=128MB" --dn-guc="bulk_write_ring_size=128MB"
--dn-guc="cstore_buffers=16MB"
Parsing the configuration file.
Check preinstall on every node.
Successfully checked preinstall on every node.
Creating the backup directory.
Successfully created the backup directory.
begin deploy..
```



```

Installing the cluster.
begin prepare Install Cluster..
Checking the installation environment on all nodes.
begin install Cluster..
Installing applications on all nodes.
Successfully installed APP.
begin init Instance..
encrypt cipher and rand files for database.
Please enter password for database:
Please repeat for database:
begin to create CA cert files
The sslcert will be generated in /opt/gaussdb/app/share/sslcert/om
Cluster installation is completed.
Configuring.
.....
Successfully started cluster.
Successfully installed application.
end deploy..

```

注意：输入 omm 用户密码时，不要输入错误（如密码：openGauss@123）。

步骤 5 清理软件安装包。

```

[omm@db1 openGauss]$ exit
logout
[root@db1 script]# cd /opt/software/openGauss/
[root@db1 openGauss]# ll
total 288M
-rwxr-xr-x. 1 omm dbgrp      1334 Jan 11 11:15 clusterconfig.xml
drwxr-xr-x. 15 root root      4096 Jan 11 11:14 lib
-rwxr-xr-x. 1 root root    99521627 Dec 31 20:58 openGauss-1.1.0-CentOS-64bit-all.tar.gz
-rwxr-xr-x. 1 root root       65 Dec 31 20:41 openGauss-1.1.0-CentOS-64bit-om.sha256
-rwxr-xr-x. 1 root root    13446137 Dec 31 20:41 openGauss-1.1.0-CentOS-64bit-om.tar.gz
-rwxr-xr-x. 1 root root       65 Dec 31 20:40 openGauss-1.1.0-CentOS-64bit.sha256
-rwxr-xr-x. 1 root root    87084796 Dec 31 20:40 openGauss-1.1.0-CentOS-64bit.tar.bz2
-rwxr-xr-x. 1 root root   100584223 Jan 11 11:14 openGauss-Package-bak_392c0438.tar.gz
drwxr-xr-x. 6 root root      4096 Dec 31 20:41 script
drwxr-xr-x. 2 root root      115 Dec 31 20:41 simpleInstall
-rwxr-xr-x. 1 root root       65 Dec 31 20:40 upgrade_sql.sha256
-rwxr-xr-x. 1 root root    134579 Dec 31 20:40 upgrade_sql.tar.gz
-rwxr-xr-x. 1 root root       32 Dec 31 20:40 version.cfg
[root@db1 openGauss]# rm -rf openGauss-1.1.0-CentOS-64bit-all.tar.gz
[root@db1 openGauss]# rm -rf openGauss-1.1.0-CentOS-64bit-om.tar.gz

```

数据库安装结束。

2 数据库使用

本节描述使用数据库的基本操作。通过此节您可以完成创建数据库、创建表及向表中插入数据和查询表中数据等操作。

2.1 前提条件

- openGauss 正常运行。
- 由于本实验是对 openGauss 数据库的基本使用，需要掌握 openGauss 数据库的基本操作和 SQL 语法，openGauss 数据库支持 SQL2003 标准语法，数据库基本操作参见附录二。

2.2 操作步骤

步骤 1 以操作系统用户 omm 登录数据库主节点。

```
[root@ecs-c9bf script]# su - omm
```

若不确定数据库主节点部署在哪台服务器，请确认连接信息。

步骤 2 启动服务。

启动服务命令：

```
[omm@ecs-c9bf ~]$ gs_om -t start
Starting cluster.
=====
=====
Successfully started.
```

步骤 3 连接数据库。

```
[omm@ecs-c9bf ~]$ 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 1.1.0 build 290d125f) compiled at 2020-05-08 02:59:43 commit 2143 last mr 131
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，请确认连接信息获取。

引申信息：

- 使用数据库前，需先使用客户端程序或工具连接到数据库，然后就可以通过客户端程序或工具执行 SQL 来使用数据库了。
- gsql 是 openGauss 数据库提供的命令行方式的数据库连接工具。

步骤 4 第一次连接数据库时，需要先修改 omm 用户密码，新密码修改为 Bigdata@123（建议用户自定义密码）。

```
postgres=# alter role omm identified by 'Bigdata@123' replace 'openGauss@123';  
ALTER ROLE
```

步骤 5 创建数据库用户。

默认只有 openGauss 安装时创建的管理员用户可以访问初始数据库，您还可以创建其他数据库用户帐号。

```
postgres=# CREATE USER joe WITH PASSWORD 'Bigdata@123';
```

当结果显示为如下信息，则表示创建成功。

```
CREATE ROLE
```

如上创建了一个用户名为 joe，密码为 Bigdata@123 的用户。

步骤 6 创建数据库。

```
postgres=# CREATE DATABASE db_tpcc OWNER joe;
```

当结果显示为如下信息，则表示创建成功。

```
CREATE DATABASE
```

创建完 db_tpcc 数据库后，就可以按如下方法退出 postgres 数据库，使用新用户连接到此数据库执行接下来的创建表等操作。当然，也可以选择继续在默认的 postgres 数据库下做后续的体验。

退出 postgres 数据库。

```
postgres=# \q
```

使用新用户连接到此数据库。

```
[omm@ecs-cgbf ~]$ gsql -d db_tpcc -p 26000 -U joe -W Bigdata@123 -r
```

当结果显示为如下信息，则表示连接成功

```
gsql ((openGauss 1.1.0 build 290d125f) compiled at 2020-05-08 02:59:43 commit 2143 last mr 131
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

db_tpcc=>
```

步骤 7 创建 SCHEMA。

```
db_tpcc=> CREATE SCHEMA joe AUTHORIZATION joe;
```

当结果显示为如下信息，则表示创建 SCHEMA 成功。

```
CREATE SCHEMA
```

步骤 8 创建表。

创建一个名称为 mytable，只有一列的表。字段名为 firstcol，字段类型为 integer。

```
db_tpcc=> CREATE TABLE mytable (firstcol int);
CREATE TABLE
```

步骤 9 向表中插入数据：

```
db_tpcc=> INSERT INTO mytable values (100);
```

当结果显示为如下信息，则表示插入数据成功。

```
INSERT 0 1
```

查看表中数据：

```
db_tpcc=> SELECT * from mytable;
 firstcol
-----
      100
(1 row)
```

步骤 10 退出 postgres 数据库。

```
postgres=# \q
```

本实验结束。

3

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。
- arguments(参数)：参数是指命令的作用对象。

3.1 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。

3.2 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。

3.3 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
```

3.4 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
```

3.5 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件包，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
```

3.6 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
```

- 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
```

3.7 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
```

3.8 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
```

3.9 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：权限设定字符串，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
```

3.10 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)。
- file：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 *
```

3.11 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
```

3.12 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)；
- -l --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。

3.13 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
```

3.14 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 testfile
```

- 把 testfile1 和 testfile2 的文档内容加上行号（空白行不加）之后将内容追加到 testfile3 文档里：

```
cat -b testfile1 testfile2 >> testfile3
```

- 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）。

4

附录二：openGauss 数据库基本操作

4.1 查看数据库对象

- 查看帮助信息：

```
postgres=# \?
```

- 切换数据库：

```
postgres=# \c dbname
```

- 列举数据库：

使用 \l 元命令查看数据库系统的数据库列表。

```
postgres=# \l
```

使用如下命令通过系统表 pg_database 查询数据库列表。

```
postgres=# SELECT datname FROM pg_database;
```

- 列举表：

```
postgres=# \dt
```

- 列举所有表、视图和索引：

```
postgres=# \d+
```

使用 gsql 的 \d+ 命令查询表的属性。

```
postgres=# \d+ tablename
```

- 查看表结构：

```
postgres=# \d tablename
```

- 列举 schema：

```
postgres=# \dn
```

- 查看索引：

```
postgres=# \di
```

- 查询表空间：

使用 gsql 程序的元命令查询表空间。

```
postgres=# \db
```


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

```
postgres=# SELECT spcname FROM pg_tablespace;
```

- 查看数据库用户列表：

```
postgres=# SELECT * FROM pg_user;
```

- 要查看用户属性：

```
postgres=# SELECT * FROM pg_authid;
```

- 查看所有角色：

```
postgres=# SELECT * FROM PG_ROLES;
```

4.2 其他操作

- 切换数据库：

```
postgres=# \c dbname
```

- 切换用户：

```
postgres=# \c - username
```

- 退出数据库：

```
postgres=# \q
```