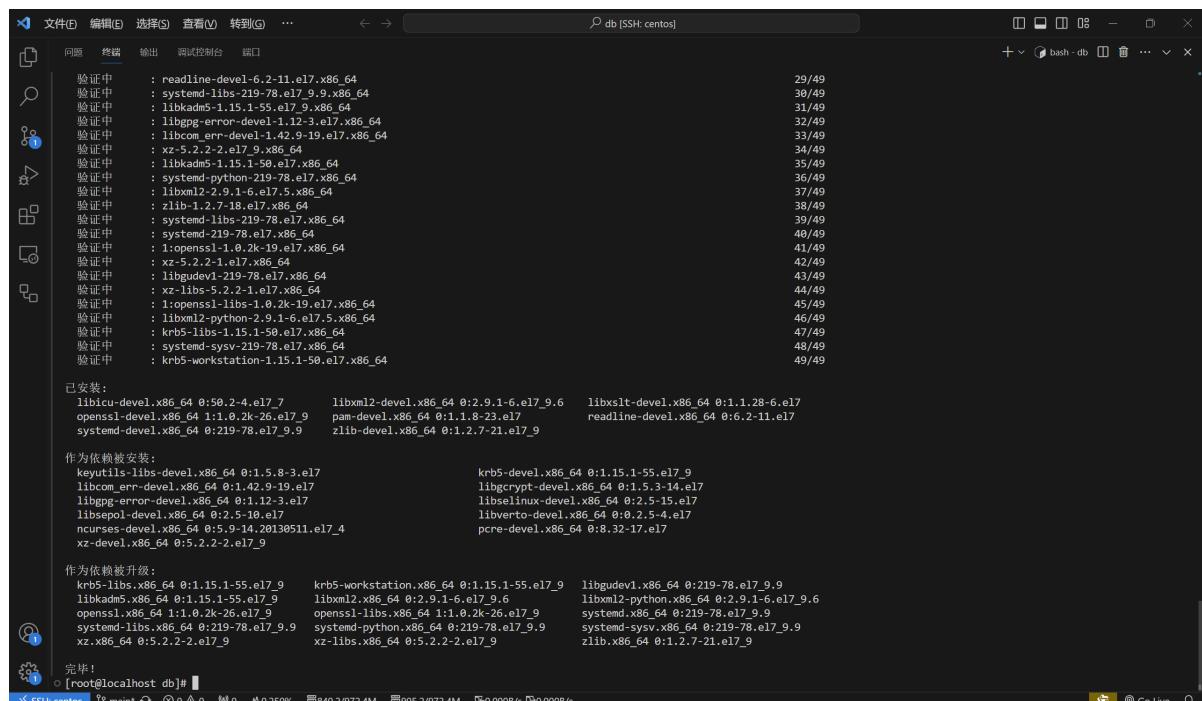


## 查看ip

```
hostname -I
```

## 安装依赖

```
sudo yum install -y readline-devel zlib-devel libicu-devel openssl-devel pam-devel libxml2-devel libxslt-devel systemd-devel
```



```
root@localhost db]# sudo yum install -y readline-devel zlib-devel libicu-devel openssl-devel pam-devel libxml2-devel libxslt-devel systemd-devel
[...]
已安装:
libicu-devel.x86_64 0:50.2-4.el7_7      libxml2-devel.x86_64 0:2.9.1-6.el7_9.6    libxslt-devel.x86_64 0:1.1.28-6.el7
openssl-devel.x86_64 1:1.0.2k-26.el7_9   pam-devel.x86_64 0:1.1.8-23.el7        readline-devel.x86_64 0:6.2-11.el7
systemd-devel.x86_64 0:219-78.el7_9     zlib-devel.x86_64 0:1.2.7-21.el7_9

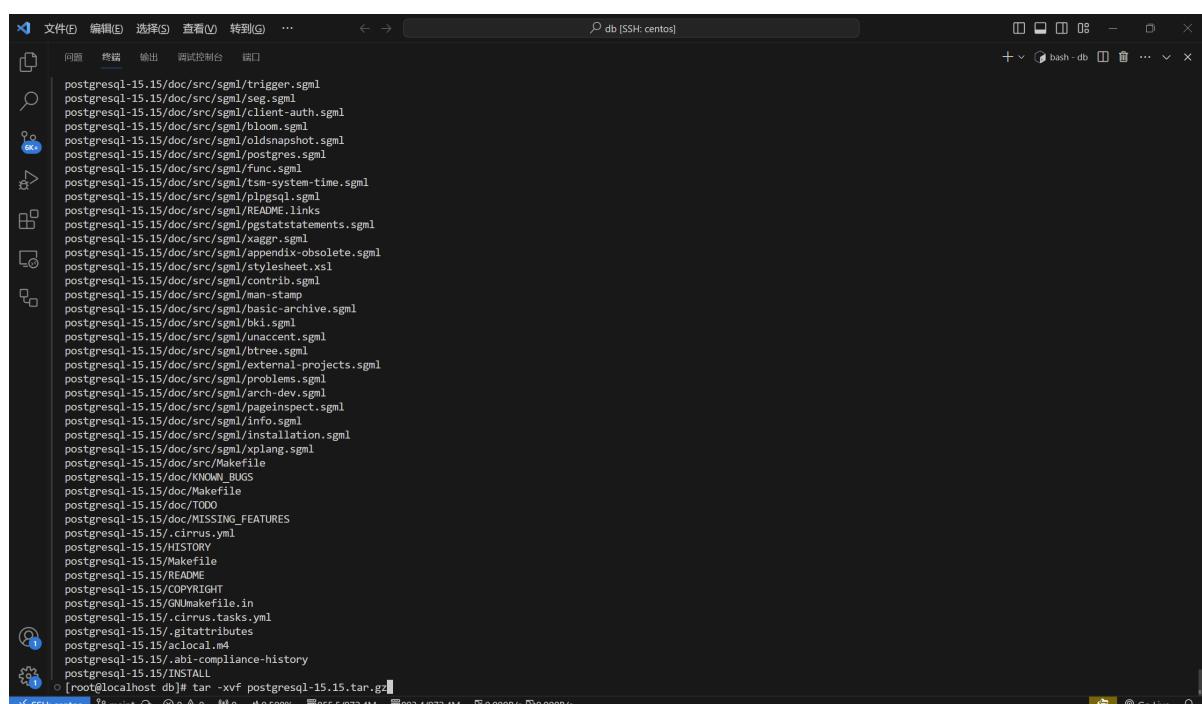
作为依赖被安装:
keyutils-libs-devel.x86_64 0:1.5.8-3.el7      krb5-devel.x86_64 0:1.15.1-55.el7_9
libcom_err-devel.x86_64 0:1.42-9.el7          libgcrypt-devel.x86_64 0:1.5.3-14.el7
libpgpg-error-devel.x86_64 0:1.12-3.el7       libselinux-devel.x86_64 0:2.5-15.el7
libsepol-devel.x86_64 0:2.5-10.el7           libverto-devel.x86_64 0:0.2.5-4.el7
ncurses-devel.x86_64 0:5.9-14.20130511.el7_4 pcre-devel.x86_64 0:8.32-17.el7
xz-devel.x86_64 0:5.2.2-2.el7_9

作为依赖被升级:
krb5-libs.x86_64 0:1.15.1-55.el7_9      krb5-workstation.x86_64 0:1.15.1-55.el7_9      libgudev1.x86_64 0:219-78.el7_9.9
libkadm5.x86_64 0:1.15.1-55.el7_9       libxml2.x86_64 0:2.9.1-6.el7_9.6      libxml2-python.x86_64 0:2.9.1-6.el7_9.6
openssl.x86_64 1:1.0.2k-26.el7_9        openssl-libs.x86_64 1:1.0.2k-26.el7_9      systemd.x86_64 0:219-78.el7_9.9
systemd-libs.x86_64 0:219-78.el7_9       systemd-python.x86_64 0:219-78.el7_9      systemd-sysv.x86_64 0:219-78.el7_9.9
xz.x86_64 0:5.2.2-2.el7_9               zlib.x86_64 0:5.2.2-2.el7_9                  zlib.x86_64 0:1.2.7-21.el7_9

完毕!
[root@localhost db]#
```

## 解压源码包并切换到目录

```
tar -xvf postgresql-15.15.tar.gz
cd postgresql-15.15/
```



```
root@localhost db]# tar -xvf postgresql-15.15.tar.gz
root@localhost db]# cd postgresql-15.15/
[...]
[root@localhost db]#
```

## 配置编译选项

```
./configure --prefix=/db/uxdb --enable-debug
```

```
checking for int64... no
checking for uint64... no
checking for __int128... yes
checking for __int128 alignment bug... ok
checking alignment of PG_INT128_TYPE... 16
checking for builtin __sync char locking functions... yes
checking for builtin __sync int32 locking functions... yes
checking for builtin __sync int32 atomic operations... yes
checking for builtin __sync int64 atomic operations... yes
checking for builtin __atomic int32 atomic operations... yes
checking for builtin __atomic int64 atomic operations... yes
checking for __get_cpuid... yes
checking for __mm_crc32_u32 and __mm_crc32_u32 with CFLAGS=... no
checking for __mm_crc32_u32 and __mm_crc32_u32 with CFLAGS=mssse4.2... yes
checking for __crc32cb, __crc32ch, __crc32cw, and __crc32cd with CFLAGS=... no
checking for __crc32cb, __crc32ch, __crc32cw, and __crc32cd with CFLAGS=march=armv8-a+crc+simd... no
checking for __crc32cb, __crc32ch, __crc32cw, and __crc32cd with CFLAGS=march=armv8-a+crc... no
checking which CRC-32C implementation to use... SSE 4.2 with runtime check
checking for library containing sem_init... none required
checking which semaphore API to use... unnamed POSIX
checking which random number source to use... /dev/urandom
checking for /dev/urandom... yes
checking for xmllint... /usr/bin/xmllint
checking for xsltproc... /usr/bin/xsltproc
checking for fop... no
checking for dbtopub... no
checking whether gcc -std=gnu99 supports -Wl,--as-needed... yes
configure: using compiler=gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-44)
configure: using CFLAGS=-Wall -Wmissing-prototypes -Wpointer-arith -Wdeclaration-after-statement -Werror=vla -Wendif-labels -Wmissing-format-attribute -Wformat-security -fno-strict-alignment -fwrapv -fexec-prefix-standard -O2
configure: using CPPFLAGS=-D_GNU_SOURCE
configure: using LDFLAGS=-Wl,--as-needed
configure: creating ./config.status
config.status: creating GNUnifakefile
config.status: creating src/Makefile_global
config.status: creating src/include/pg_config.h
config.status: creating src/include/pg_config_ext.h
config.status: creating src/interfaces/ecpg/include/ecpg_config.h
config.status: linking src/backend/port/tas_dummy.s to src/backend/port/tas.s
config.status: linking src/backend/port/posix_sema.c to src/backend/port/pg_sema.c
config.status: linking src/backend/port/sys_shmem.c to src/backend/port/pg_shmem.c
config.status: linking src/include/port/linux.h to src/include/pg_config_os.h
config.status: linking src/makefiles/Makefile.linux to src/Makefile_port
[root@localhost postgresql-15.15]#
```

## 编译与安装

```
sudo make && sudo make install
```

```
文件(E) 编辑(E) 选择(S) 查看(V) 转到(G) ... ← → db [SSH: centos]
问题 终端 输出 调试控制台 端口 + × bash - postgresql-15.15

/usr/bin/install -c pg_regress '/usr/local/pgsql/lib/pgxs/src/test/regress/pg_regress'
make[2]: 离开目录 "/root/db/postgresql-15.15/src/test/regress"
make -C test/isolation install
make[2]: 进入目录 "/root/db/postgresql-15.15/src/test/isolation"
make -C ../../src/interfaces/libpq all
make[3]: 进入目录 "/root/db/postgresql-15.15/src/interfaces/libpq"
make -C ../../src/port all
make[4]: 进入目录 "/root/db/postgresql-15.15/src/port"
make[4]: 对 "all" 无需做任何事。
make[4]: 离开目录 "/root/db/postgresql-15.15/src/port"
make -C ../../src/common all
make[4]: 进入目录 "/root/db/postgresql-15.15/src/common"
make[4]: 对 "all" 无需做任何事。
make[4]: 离开目录 "/root/db/postgresql-15.15/src/common"
make[3]: 离开目录 "/root/db/postgresql-15.15/src/interfaces/libpq"
make -C ../../src/port all
make[3]: 进入目录 "/root/db/postgresql-15.15/src/port"
make[3]: 对 "all" 无需做任何事。
make[3]: 离开目录 "/root/db/postgresql-15.15/src/port"
make -C ../../src/common all
make[3]: 进入目录 "/root/db/postgresql-15.15/src/common"
make[3]: 对 "all" 无需做任何事。
make[3]: 离开目录 "/root/db/postgresql-15.15/src/common"
/usr/bin/mkdir -p '/usr/local/pgsql/lib/pgxs/src/test/isolation'
/usr/bin/install -c pg_isolation_regress '/usr/local/pgsql/lib/pgxs/src/test/isolation/pg_isolation_regress'
/usr/bin/install -c isolationtester '/usr/local/pgsql/lib/pgxs/src/test/isolation/isolationtester'
make[2]: 离开目录 "/root/db/postgresql-15.15/src/test/isolation"
make -C test/perl install
make[2]: 进入目录 "/root/db/postgresql-15.15/src/test/perl"
make[2]: 对 "install" 无需做任何事。
make[2]: 离开目录 "/root/db/postgresql-15.15/src/test/perl"
/usr/bin/mkdir -p '/usr/local/pgsql/lib/pgxs/src'
/usr/bin/install -c -m 644 Makefile.global '/usr/local/pgsql/lib/pgxs/src/Makefile.global'
/usr/bin/install -c -m 644 Makefile.port '/usr/local/pgsql/lib/pgxs/src/Makefile.port'
/usr/bin/install -c -m 644 Makefile.shlib '/usr/local/pgsql/lib/pgxs/src/Makefile.shlib'
/usr/bin/install -c -m 644 ./nls-global.mk '/usr/local/pgsql/lib/pgxs/src/nls-global.mk'
make[2]: 离开目录 "/root/db/postgresql-15.15/src"
make -C config install
make[1]: 进入目录 "/root/db/postgresql-15.15/config"
/usr/bin/mkdir -p '/usr/local/pgsql/config'
/usr/bin/install -c -m 755 ./install-sh '/usr/local/pgsql/lib/pgxs/config/install-sh'
/usr/bin/install -c -m 755 ./missing '/usr/local/pgsql/lib/pgxs/config/missing'
make[1]: 离开目录 "/root/db/postgresql-15.15/config"

① [root@localhost postgresql-15.15]#
```

# 安装插件

```
cd contrib/pageinspect/  
sudo make && sudo make instal
```

```
文件(B) 编辑(E) 选择(U) 查看(V) 转到(G) ... ← → db [SSH: centos]
问题 终端 输出 调试控制台 端口 + bash -pageinspect
make[2]: 离开目录 "/root/db/postgresql-15.15/src/backend/utils"
make[1]: 离开目录 "/root/db/postgresql-15.15/src/backend"
gcc -std=gnu99 -Wall -Wmissing-prototypes -Wpointer-arith -Wdeclaration-after-statement -Werror=vla -Wendif-labels -Wmissing-format-attribute -Wformat-security -fno-strict-aliasing -fwrapv -fexec-precision-standard -O2 -fPIE -I . -I ./src/include -D GNU_SOURCE -c -o brinfuncs.o brinfuncs.c
gcc -std=gnu99 -Wall -Wmissing-prototypes -Wpointer-arith -Wdeclaration-after-statement -Werror=vla -Wendif-labels -Wmissing-format-attribute -Wformat-security -fno-strict-aliasing -fwrapv -fexec-precision-standard -O2 -fPIE -I . -I ./src/include -D GNU_SOURCE -c -o btreefuncs.o btreefuncs.c
gcc -std=gnu99 -Wall -Wmissing-prototypes -Wpointer-arith -Wdeclaration-after-statement -Werror=vla -Wendif-labels -Wmissing-format-attribute -Wformat-security -fno-strict-aliasing -fwrapv -fexec-precision-standard -O2 -fPIE -I . -I ./src/include -D GNU_SOURCE -c -o fsmfuncs.o fsmfuncs.c
gcc -std=gnu99 -Wall -Wmissing-prototypes -Wpointer-arith -Wdeclaration-after-statement -Werror=vla -Wendif-labels -Wmissing-format-attribute -Wformat-security -fno-strict-aliasing -fwrapv -fexec-precision-standard -O2 -fPIE -I . -I ./src/include -D GNU_SOURCE -c -o ginfuncs.o ginfuncs.c
gcc -std=gnu99 -Wall -Wmissing-prototypes -Wpointer-arith -Wdeclaration-after-statement -Werror=vla -Wendif-labels -Wmissing-format-attribute -Wformat-security -fno-strict-aliasing -fwrapv -fexec-precision-standard -O2 -fPIE -I . -I ./src/include -D GNU_SOURCE -c -o hashfuncs.o hashfuncs.c
gcc -std=gnu99 -Wall -Wmissing-prototypes -Wpointer-arith -Wdeclaration-after-statement -Werror=vla -Wendif-labels -Wmissing-format-attribute -Wformat-security -fno-strict-aliasing -fwrapv -fexec-precision-standard -O2 -fPIE -I . -I ./src/include -D GNU_SOURCE -c -o heapfuncs.o heapfuncs.c
gcc -std=gnu99 -Wall -Wmissing-prototypes -Wpointer-arith -Wdeclaration-after-statement -Werror=vla -Wendif-labels -Wmissing-format-attribute -Wformat-security -fno-strict-aliasing -fwrapv -fexec-precision-standard -O2 -fPIE -I . -I ./src/include -D GNU_SOURCE -c -o rawpage.o rawpage.c
gcc -std=gnu99 -Wall -Wmissing-prototypes -Wpointer-arith -Wdeclaration-after-statement -Werror=vla -Wendif-labels -Wmissing-format-attribute -Wformat-security -fno-strict-aliasing -fwrapv -fexec-precision-standard -O2 -fPIE -I . -I ./src/include -D GNU_SOURCE -c -o rawpage_label.o rawpage_label.c
[src]/root/localhost/pageinspect]$ make install
make: C ./src/backend/generated-headers
make[1]: 进入目录 "/root/db/postgresql-15.15/src/backend"
make[1]: 对 catalog distprep generated-header-symlinks
make[2]: 进入目录 "/root/db/postgresql-15.15/src/backend/catalog"
make[2]: 对 "distprep" 无须做任何事。
make[2]: 对 "generated-header-symlinks" 无须做任何事。
make[2]: 离开目录 "/root/db/postgresql-15.15/src/backend/catalog"
make[1]: 对 catalog distprep generated-header-symlinks
make[2]: 进入目录 "/root/db/postgresql-15.15/src/backend/utils"
make[2]: 对 "distprep" 无须做任何事。
make[2]: 对 "generated-header-symlinks" 无须做任何事。
make[2]: 离开目录 "/root/db/postgresql-15.15/src/backend/utils"
make[1]: 离开目录 "/root/db/postgresql-15.15/src/backend"
/usr/bin/mkdir -p '/usr/local/pgsql/lib'
/usr/bin/mkdir -p '/usr/local/pgsql/share/extension'
/usr/bin/mkdir -p '/usr/local/pgsql/share/extension'
/usr/bin/install -c -m 755 pageinspect.so '/usr/local/pgsql/lib/pageinspect.so'
/usr/bin/install -c -m 644 ./pageinspect.control '/usr/local/pgsql/share/extension'
/usr/bin/install -c -m 644 ./pageinspect-1.10--1.11.sql ./pageinspect-1.9--1.10.sql ./pageinspect--1.8--1.9.sql ./pageinspect--1.7--1.8.sql ./pageinspect--1.6--1.7.sql ./pageinspect--1.5.sql ./pageinspect-1.5--1.6.sql ./pageinspect-1.4--1.5.sql ./pageinspect--1.3--1.4.sql ./pageinspect--1.2--1.3.sql ./pageinspect--1.1--1.2.sql ./pageinspect--1.0--1.1.sql '/usr/local/pgsql/share/extension'
[src]/root/localhost/pageinspect]$
```

添加用户

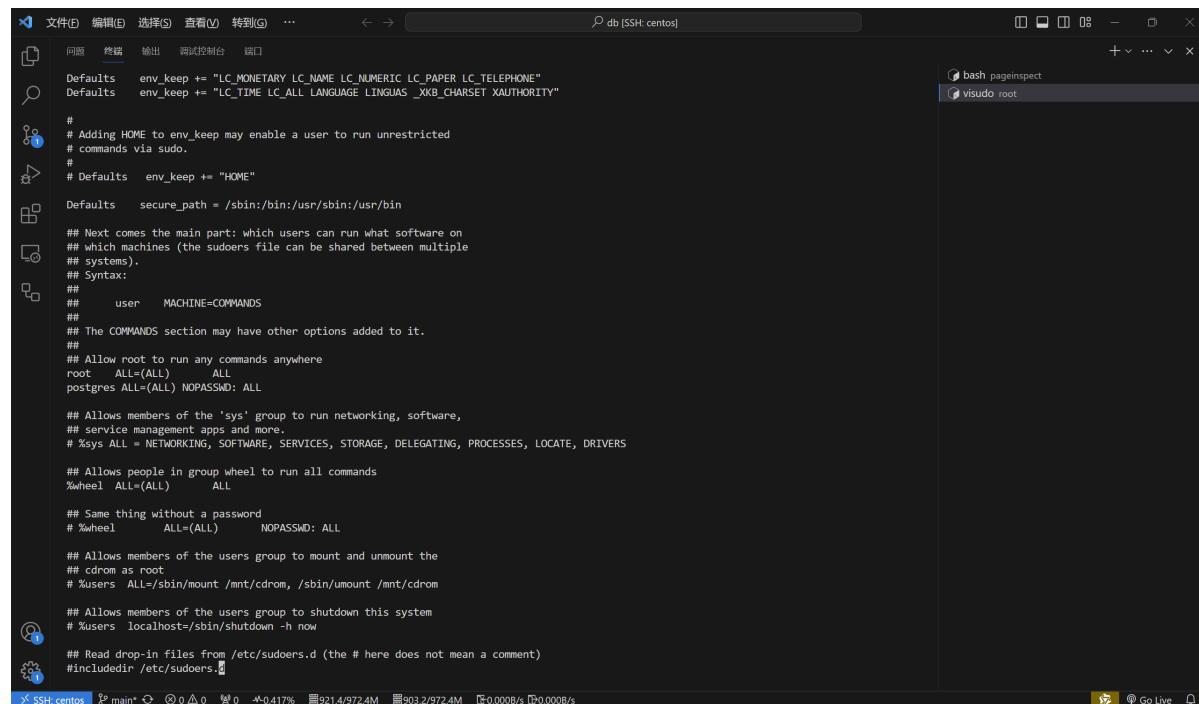
```
sudo adduser uxdb
```

# 使用 visudo 命令编辑配置文件

```
visudo
```

## 授予 uxdb 用户完整的 sudo 权限且无需输入密码

```
uxdb ALL=(ALL) NOPASSWD: ALL
```



```
Defaults env_keep += "LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE"
Defaults env_keep += "LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY"

#
# Adding HOME to env_keep may enable a user to run unrestricted
# commands via sudo.
#
# Defaults env_keep += "HOME"

Defaults secure_path = /sbin:/bin:/usr/sbin:/usr/bin

## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##     user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)      ALL
postgres ALL=(ALL) NOPASSWD: ALL

## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)      ALL

## Same thing without a password
# %wheel    ALL=(ALL)      NOPASSWD: ALL

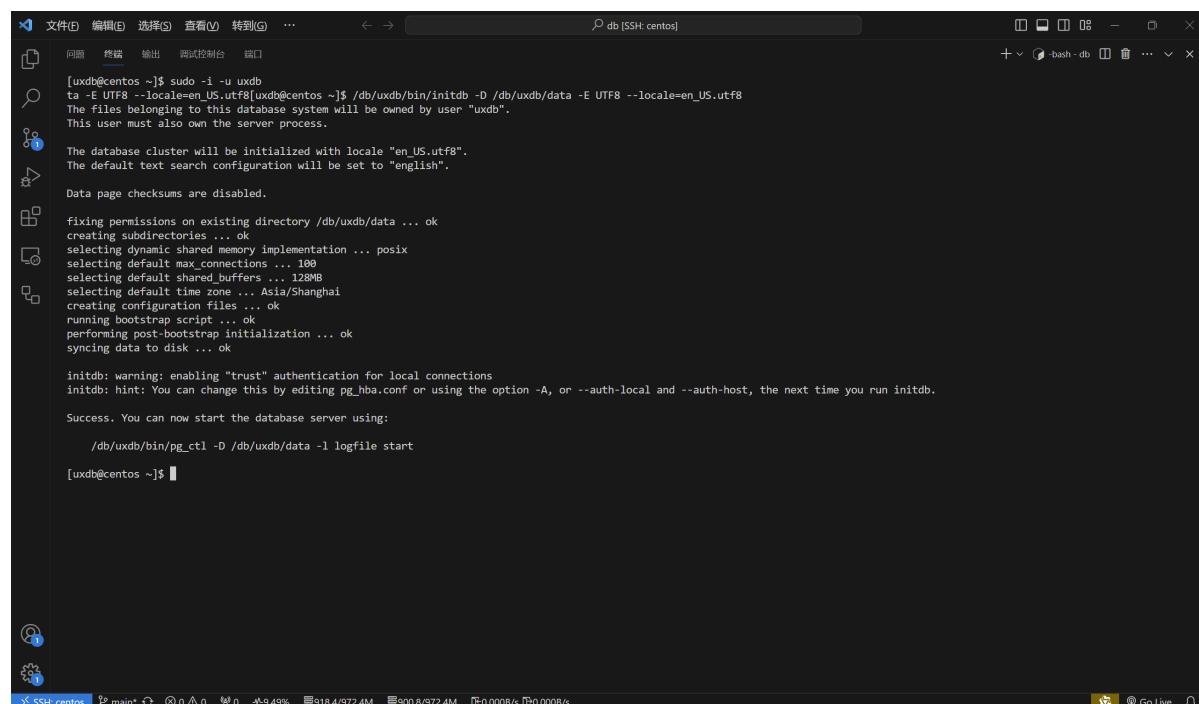
## Allows members of the users group to mount and umount the
## cdrom as root
# %users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
# %users  localhost=/sbin/shutdown -h now

## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
#includedir /etc/sudoers.d
```

## 切换到 uxdb 用户

```
sudo chmod -R 777 /db
sudo -i -u uxdb
```



```
[uxdb@centos ~]$ sudo -i -u uxdb
[uxdb@centos ~]$ /db/uxdb/bin/initdb -E UTF8 --locale=en_US.utf8
[uxdb@centos ~]$ /db/uxdb/bin/initdb -E UTF8 --locale=en_US.utf8
The files belonging to this database system will be owned by user "uxdb".
This user must also own the server process.

The database cluster will be initialized with locale "en_US.utf8".
The default text search configuration will be set to "english".

Data page checksums are disabled.

fixing permissions on existing directory /db/uxdb/data ... ok
creating subdirectories ... ok
selecting dynamic shared memory implementation ... posix
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting default time zone ... Asia/Shanghai
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok

initdb: warning: enabling "trust" authentication for local connections
initdb: hint: You can change this by editing pg_hba.conf or using the option -A, or --auth-local and --auth-host, the next time you run initdb.

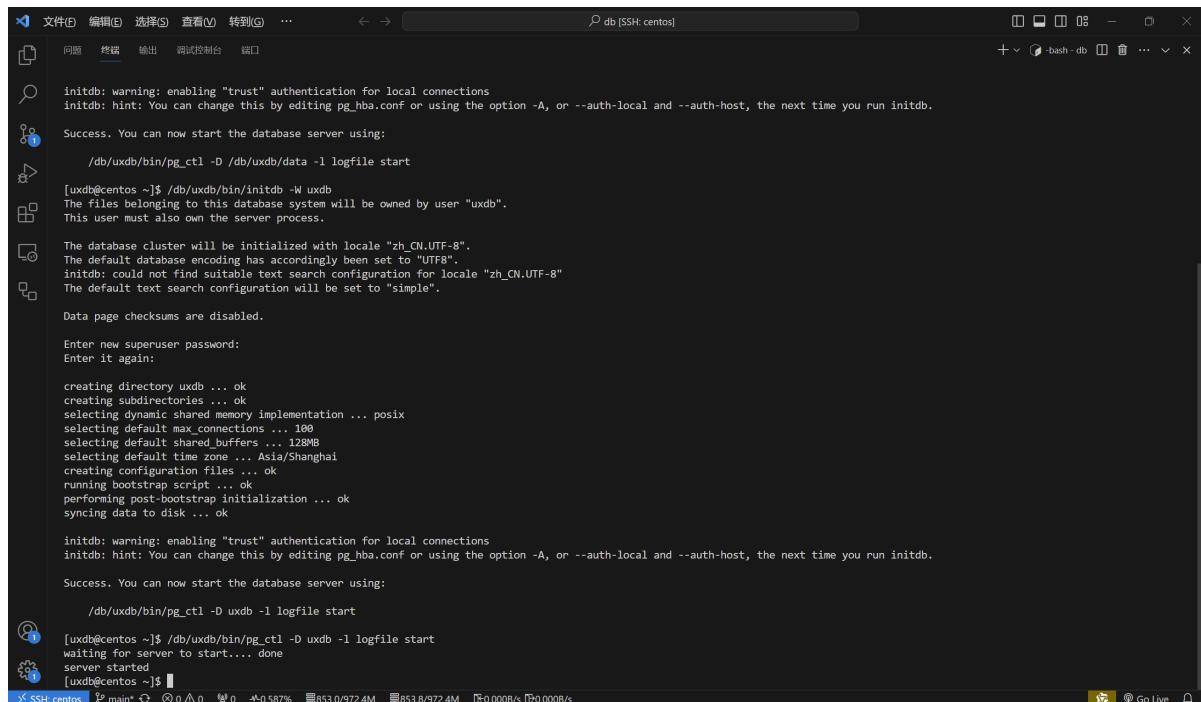
Success. You can now start the database server using:

    /db/uxdb/bin/pg_ctl -D /db/uxdb/data -l logfile start

[uxdb@centos ~]$
```

# 初始化数据库实例和启动数据库实例

```
/db/uxdb/bin/initdb -w uxdb  
/db/uxdb/bin/pg_ctl -D uxdb -l logfile start
```



```
initdb: warning: enabling "trust" authentication for local connections  
initdb: hint: You can change this by editing pg_hba.conf or using the option -A, or --auth-local and --auth-host, the next time you run initdb.  
Success. You can now start the database server using:  
/db/uxdb/bin/pg_ctl -D /db/uxdb/data -l logfile start  
[uxdb@centos ~]$ /db/uxdb/bin/initdb -W uxdb  
The files belonging to this database system will be owned by user "uxdb".  
This user must also own the server process.  
The database cluster will be initialized with locale "zh_CN.UTF-8".  
The default database encoding has accordingly been set to "UTF8".  
initdb: could not find suitable text search configuration for locale "zh_CN.UTF-8"  
The default text search configuration will be set to "simple".  
Data page checksums are disabled.  
Enter new superuser password:  
Enter it again:  
creating directory uxdb ... ok  
creating subdirectories ... ok  
selecting dynamic shared memory implementation ... posix  
selecting default max_connections ... 100  
selecting default shared_buffers ... 128MB  
selecting default time zone ... Asia/Shanghai  
creating configuration file ... ok  
running bootstrap script ... ok  
performing post-bootstrap initialization ... ok  
syncing data to disk ... ok  
initdb: warning: enabling "trust" authentication for local connections  
initdb: hint: You can change this by editing pg_hba.conf or using the option -A, or --auth-local and --auth-host, the next time you run initdb.  
Success. You can now start the database server using:  
/db/uxdb/bin/pg_ctl -D uxdb -l logfile start  
[uxdb@centos ~]$ /db/uxdb/bin/pg_ctl -D uxdb -l logfile start  
waiting for server to start... done  
server started  
[uxdb@centos ~]$
```

## 修改环境变量

```
sudo vi /etc/profile
```

在文件的末尾添加以下行:

```
export PATH=/db/uxdb/bin:$PATH
```

保存并生效: 保存文件后, 使用 source命令使配置立即在当前终端生效, 或者注销后重新登录。

```
source /etc/profile
```