**Oracle\_11g安装在RedHat6.5 （64位）**

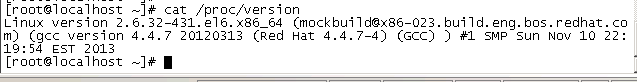
# 环境准备

Linux服务器：Redhat 6.5 64位

oracle服务器：oracle11g 64位

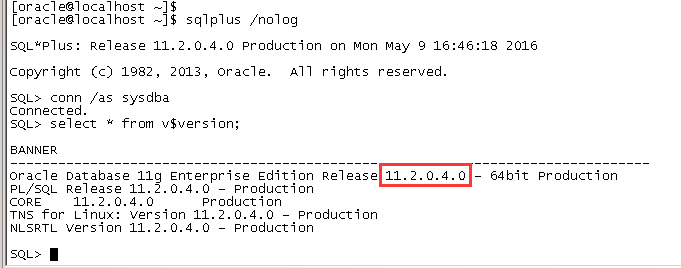
查看Linux服务器版本：

[root@localhost ~]# cat /proc/version



查看oracle服务器版本（oracle已安装完毕条件下查看）：

SQL> select \* from v$version;



请务必保证oracle版本一致（包括小版本也要一致）

# Oracle安装包准备

p13390677\_112040\_Linux-x86-64\_1of7.zip

p13390677\_112040\_Linux-x86-64\_2of7.zip

在windows环境两个包同时解压，解压后生成一个database目录，将该目录上传到linux服务器。

也可以先将两个压缩包上传到linux服务器，然后在服务器目录下解压，解压命令：

unzip p13390677\_112040\_Linux-x86-64\_1of7.zip

unzip p13390677\_112040\_Linux-x86-64\_2of7.zip

上传oracle安装包可以使用root用户（需事先创建相应的目录），且上传后需要更改安装包的所属用户：

[root@localhost ~]#chown -R oracle:oinstall /home/oracle/database

也可等创建完oracle用户后使用oracle用户上传（本手册使用这种方式）。

# Linux服务器系统配置

使用root用户进行系统配置

## 关闭SELinux、防火墙

关闭防火墙使得数据库1521端口打开，后期部署应用时用的7001、8001等端口也打开。如果需要打开防火墙就得开放上述端口以及允许ip通过。

关闭防火墙：

[root@localhost ~]#service iptables stop

[root@localhost ~]#chkconfig iptables off

关闭SELinux：

[root@localhost ~]#vi /etc/selinux/config

把SELINUX=enforcing 改为：SELINUX=disabled ，重启linux服务器生效。

重启linux服务器：

[root@localhost ~]#reboot

检查SELinux状态：

[root@localhost ~]#getenforce

**centos 7.2关闭防火墙**：

firewall-cmd --state #查看默认防火墙状态（关闭后显示notrunning，开启后显示running）

systemctl stop firewalld.service #停止firewall

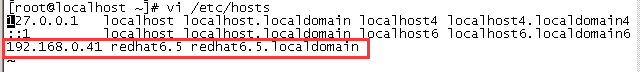
systemctl disable firewalld.service #禁止firewall开机启动

init 6 #重启虚拟机，然后再输入第一条命令查看防火墙状态

## 检查hosts文件

[root@localhost ~]# vi /etc/hosts

增加linux服务器的IP和主机名映射



## 修改linux内核

[root@localhost ~]# vi /etc/sysctl.conf

增加以下内容：

fs.suid\_dumpable = 1

fs.aio-max-nr = 1048576

fs.file-max = 6815744

kernel.shmall = 2097152

kernel.shmmax = 536870912

kernel.shmmni = 4096

kernel.sem = 250 32000 100 128

net.ipv4.ip\_local\_port\_range = 9000 65500

net.core.rmem\_default=4194304

net.core.rmem\_max=4194304

net.core.wmem\_default=262144

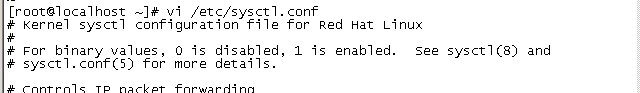
net.core.wmem\_max=1048586

保存：

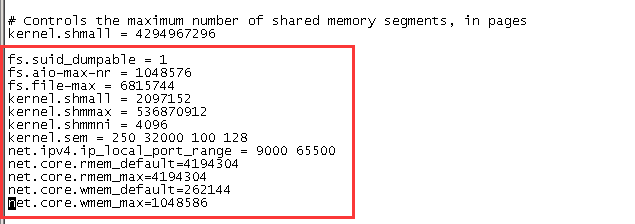
:wq

使修改生效：

[root@localhost ~]# sysctl -p



……



## 修改用户的SHELL限制

[root@localhost ~]# vi /etc/security/limits.conf

增加以下内容：

oracle soft nproc 2047

oracle hard nproc 16384

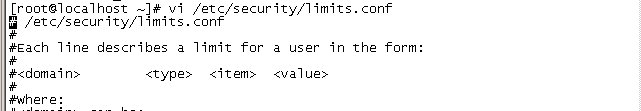
oracle soft nofile 4096

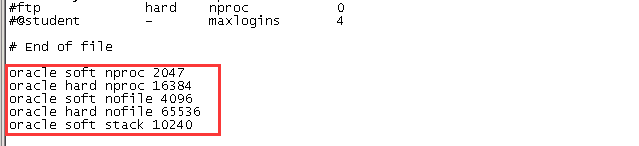
oracle hard nofile 65536

oracle soft stack 10240

保存：

:wq





## 修改/etc/pam.d/login

[root@localhost ~]# vi /etc/pam.d/login

增加以下内容：

session required /lib64/security/pam\_limits.so

session required pam\_limits.so

（我在虚拟机上部署时，linux服务器的/lib/security目录下为空，所以没有配置该步骤，后期安装oracle并没有出现问题）

## 修改/etc/profile

[root@localhost ~]# vi /etc/profile

增加以下内容：

if [ $USER = "oracle" ]; then

if [ $SHELL = "/bin/ksh" ]; then

ulimit -p 16384

ulimit -n 65536

else

ulimit -u 16384 -n 65536

fi

fi

（我在虚拟机上部署时，也没有执行该步骤，后期安装oracle并没有出现问题）

## 检查并安装oracle依赖的rpm包

**需要的rpm包如下：**

binutils- （无影响）

compat-libstdc++

gcc

gcc-c++

glibc

glibc-32bit (32 bit)

glibc-devel

glibc-devel-32bit (32 bit)

libaio

libaio-32bit (32 bit)

libaio-devel

libgcc

libstdc++

libstdc++-devel

make

sysstat

elfutils-libelf-devel

pdksh（无影响）

**检查rpm包是否安装（以binutils包为例）：**

[root@localhost ~]# rpm -q binutils



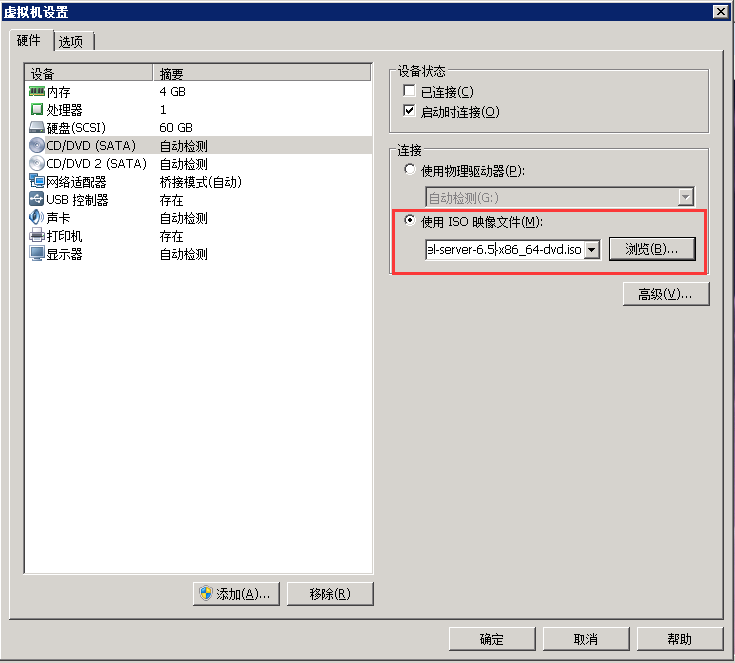
如图查到该包相应的版本则说明已安装，如查询结果为“package xxx is not installed”，则需要手动安装，上述包一个一个检查。

注：其实：还有一种更简单的检查方法，即在oracle 安装过程中会有一项是检测安装环境，到时候缺少什么包，便安装什么包即可（后面安装数据库界面会说明）

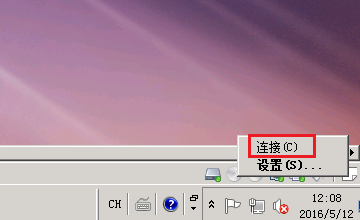
**挂载光盘并安装rmp包：（远程服务器安装时此步可忽略）**

安装rpm包之前需先挂载系统ISO安装盘，虚拟机挂盘步骤如下：

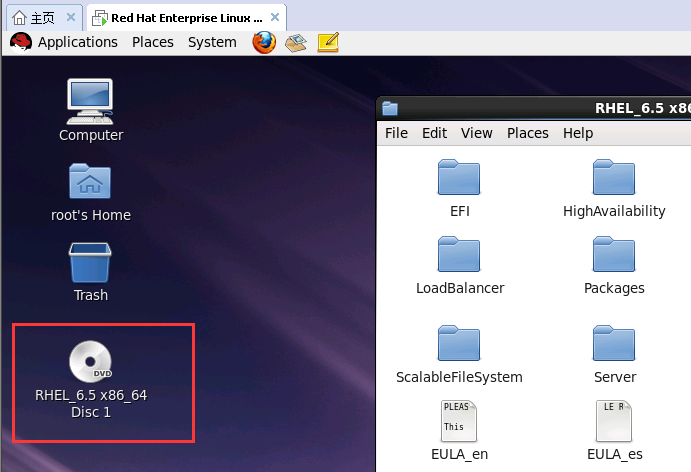
1. 进入“虚拟机->设置->硬件->CD/DVD(STAT)->连接”，选择“使用ISO映像文件”并选择ISO文件路径，确定；



1. 右键点击虚拟机界面右下角的光盘图标，点击连接完成挂载；



1. 挂载完成后桌面出现该光盘的图标



**安装rpm包：**

1. 进入Package目录

[root@localhost ~]# cd /media/RHEL\_6.5\ x86\_64\ Disc\ 1/Packages/

1. 根据上述检查不存在的包一个一个进行安装（以binutils为例）

[root@localhost Packages]# yum install binutils-2.20.51.0.2-5.36.el6.x86\_64.rpm

根据提示一步一步安装完毕。

注：检查rpm包时无需关注其版本号，安装rpm包时则需带版本号；

安装时如出现提示需先安装XX包，则根据提示先安装所缺的包；

（我在安装rpm包时pdksh这个包找不到，所以没有安装，后面安装oracle时监测勾选ignore后继续。）

# 创建oracle组和用户

使用root用户创建

## 创建组

创建oracle安装组oinstall和数据库管理组dba:

[root@localhost ~]# groupadd -g 601 dba

[root@localhost ~]# groupadd -g 602 oinstall

## 创建oracle用户

[root@localhost ~]#useradd -u 611 -m -g oinstall -G dba -d /home/oracle oracle

设置密码：

[root@localhost ~]# passwd oracle

# 上传oracle安装包

使用oracle用户上传和操作

1. 创建安装包放置目录：

[oracle@localhost ~]$mkdir /home/oracle/package

1. 使用SecureFXPortable工具上传安装包到该目录；
2. 为database目录及其里面的文件赋予执行权限：

[oracle@localhost ~]$chmod +x -R /home/oracle/package/database

# 安装oracle数据库

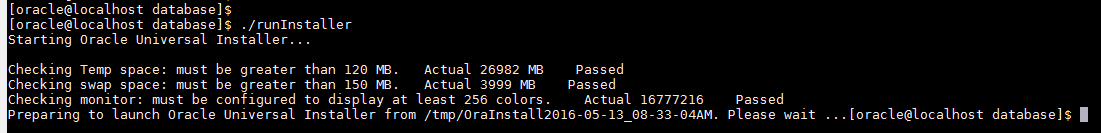
使用oracle用户操作

1. **安装oracle数据库**

界面安装时推荐使用MobaXterm\_Personal\_6.6.exe工具连接。

进入database文件夹中，cd /home/oracle/package/database(具体看上传的路径)

进行安装，输入 ./runInstaller.sh弹出安装界面



**可能遇到的问题：**

1. 出现DISPLAY之类的错误：

（使用MobaXterm\_Personal\_6.6.exe可避免此错误）

1. 将系统注销，切换oracle用户登录（并不是su - oracle）

进入database文件夹中，***cd /opt/oracle/database*** (具体看上传的路径)

进行安装，输入 **./runInstaller.sh** 弹出安装界面

1. 更换一台纯屏的显示器（一般不需要B方案）

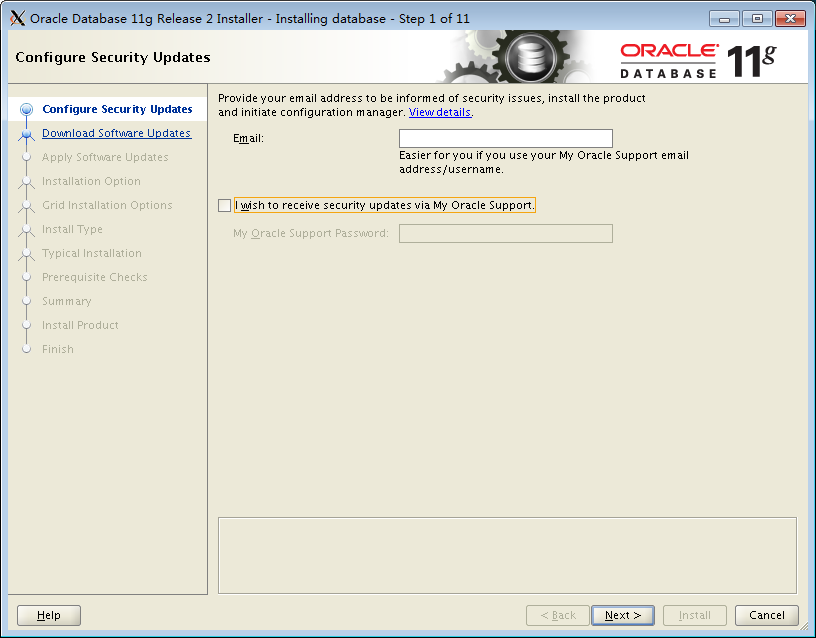
重启后，先切换到root用户，执行xhost +

然后再切换到oracle用户，执行export DISPLAY=localhost:1

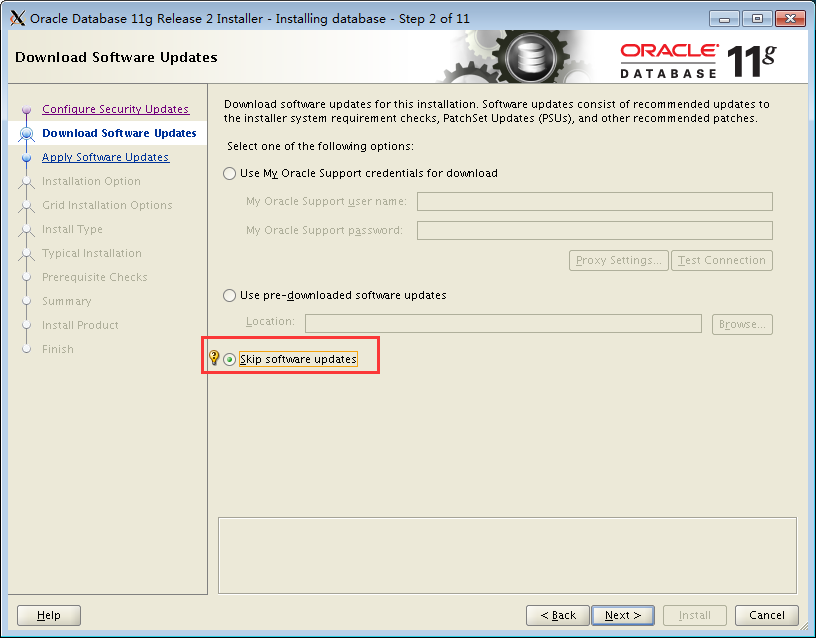
然后再进行安装

b、若打开oracle安装界面出现乱码，在安装之前输入：***export LC\_CTYPE=en\_US.UTF-8***后在安装

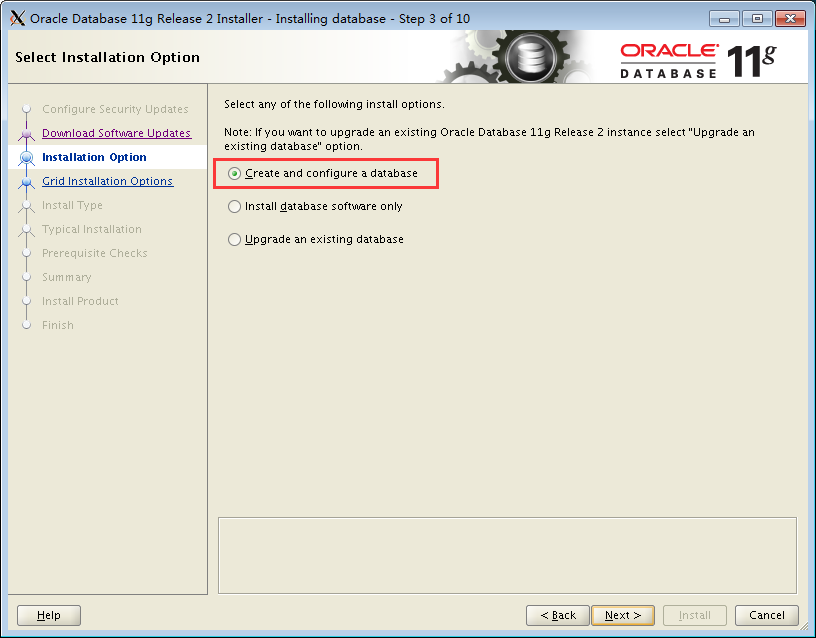
1. **oracle界面安装步骤**

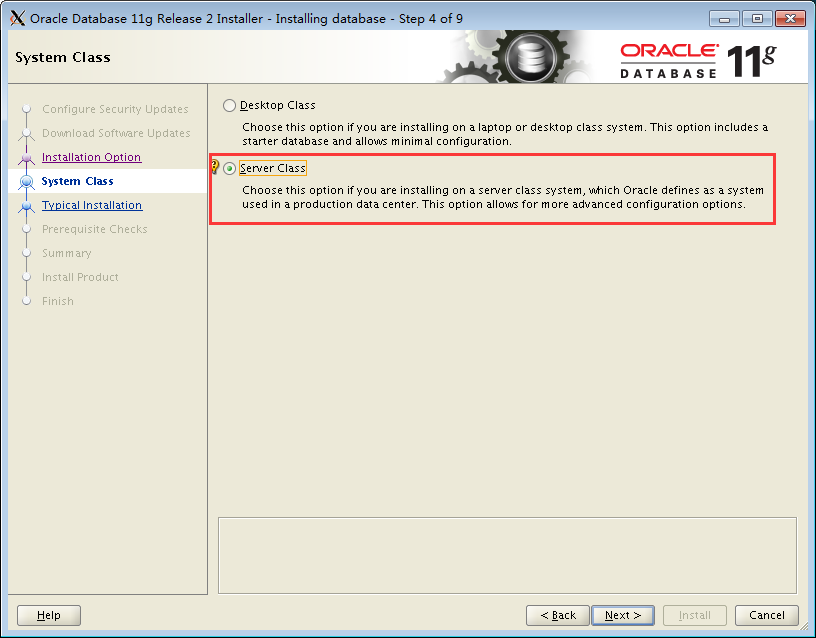


**注：一般不用勾选，直接下一步，警告直接点yes**

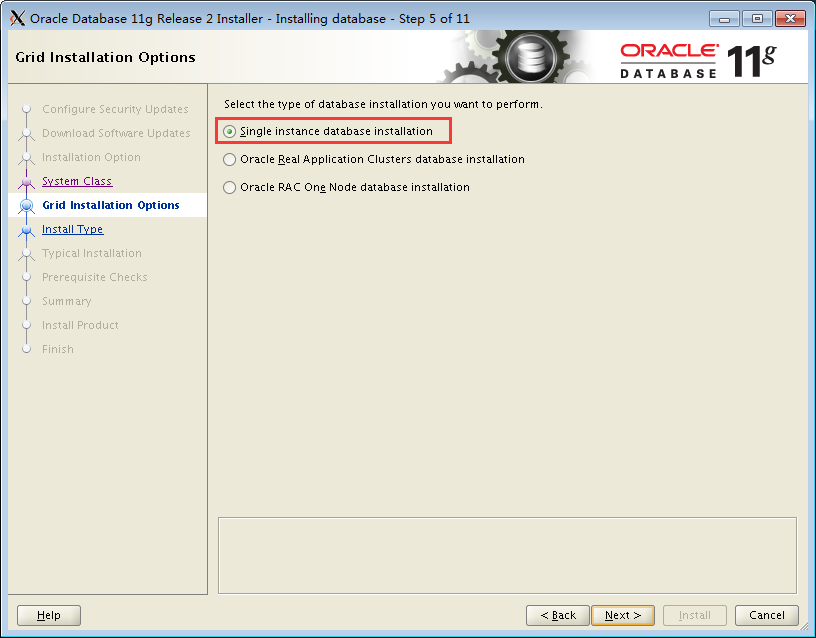


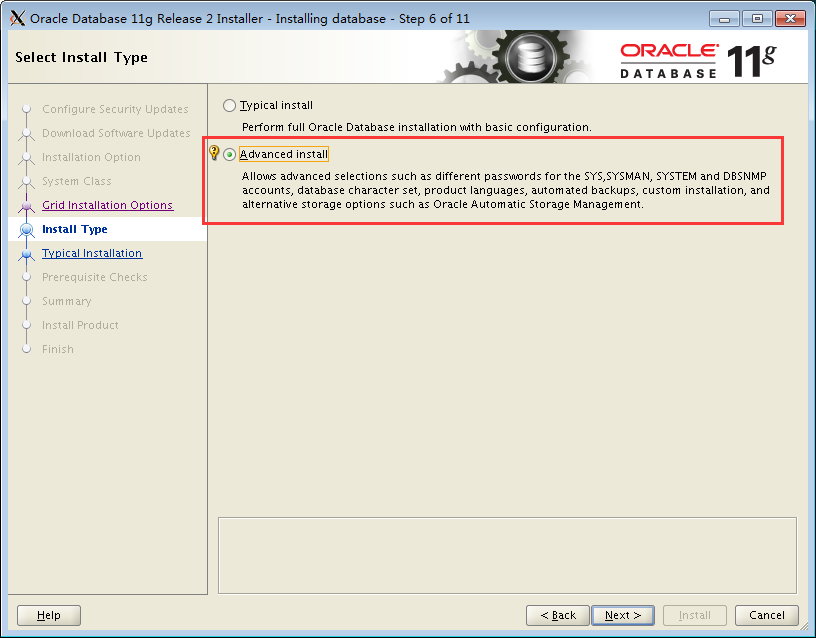
**如有推荐升级的提示，直接忽略。**



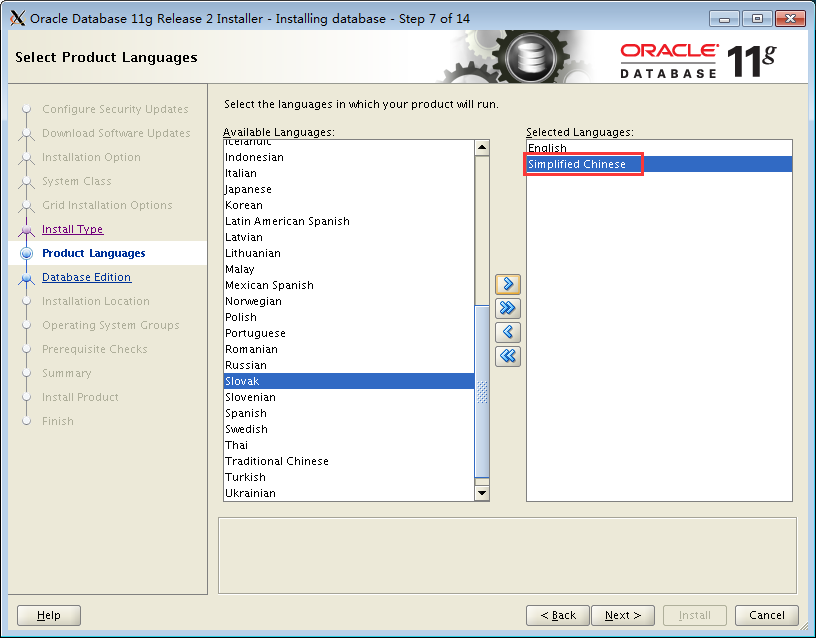


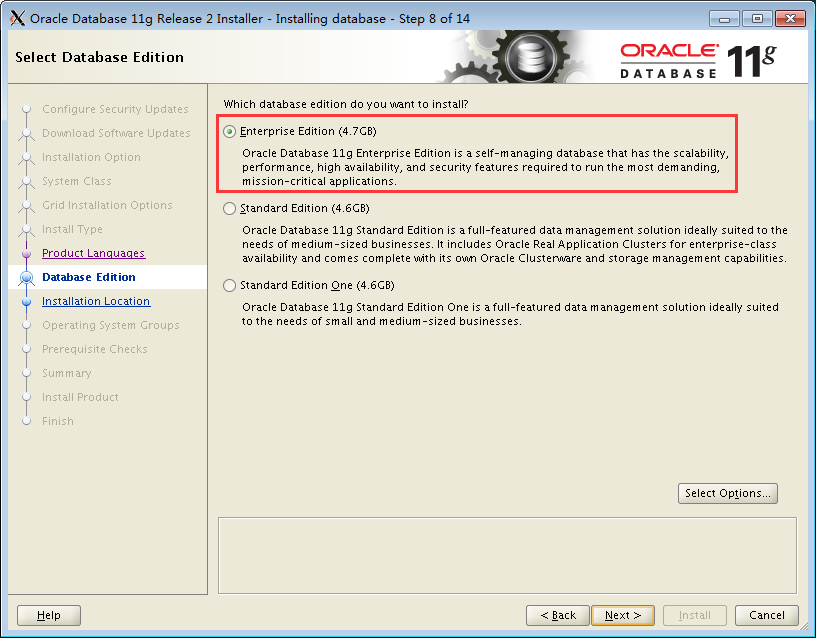
**注：此处选择服务类型的**

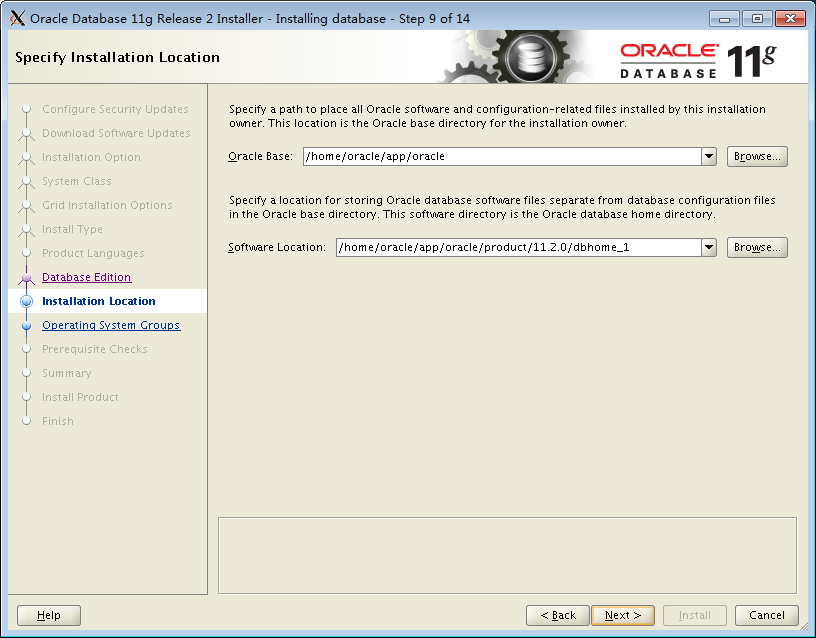




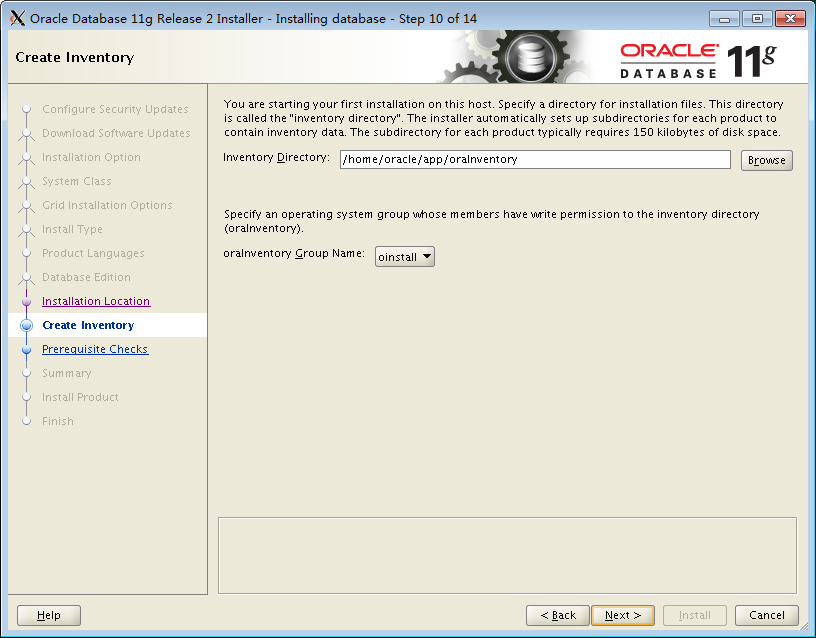
**注：此处我们进行高级安装，因为后面需要对字符集进行设置**

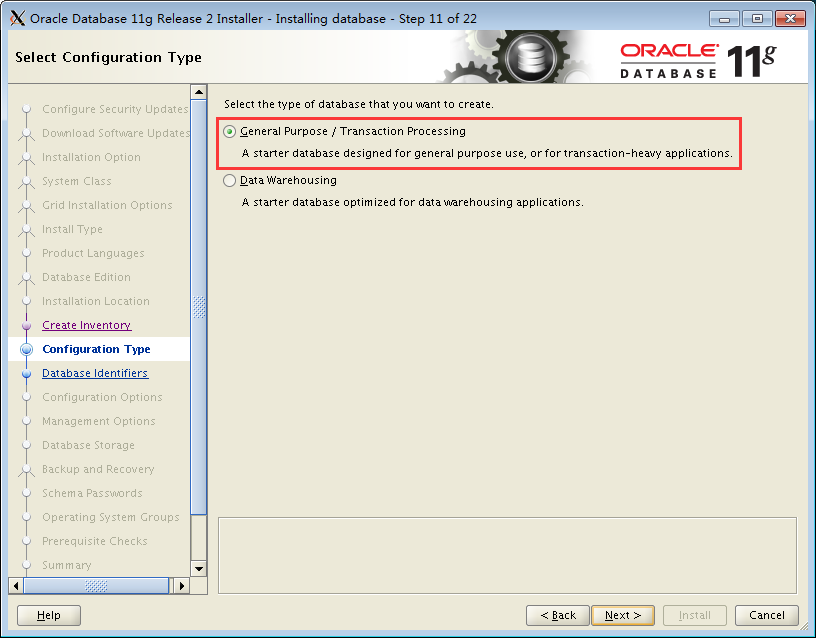


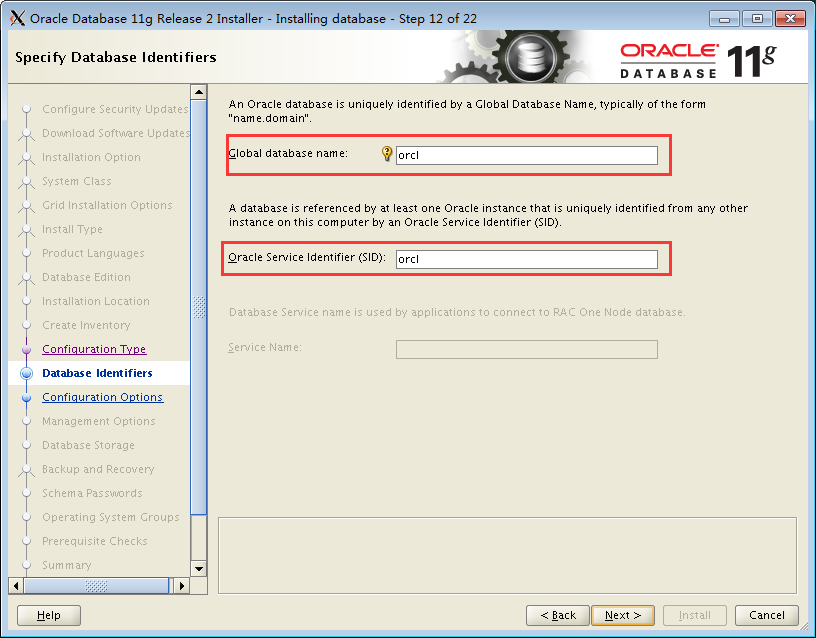




**路径自己选好，一般默认就好，不需修改**







**注：此处的配置会涉及到后期使用PL/SQL时，oracle的tnsnames.ora配置问题，配置中的SERVICE\_NAME与此处设置有关**

**如：**

**ORCL =**

**(DESCRIPTION =**

**(ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.1.100)(PORT = 1521))**

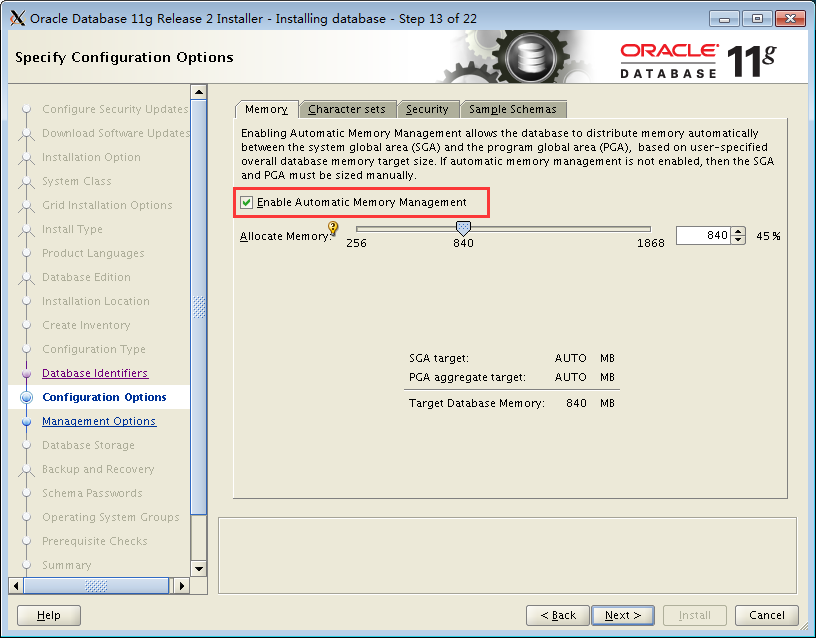
**(CONNECT\_DATA =**

**(SERVER = DEDICATED)**

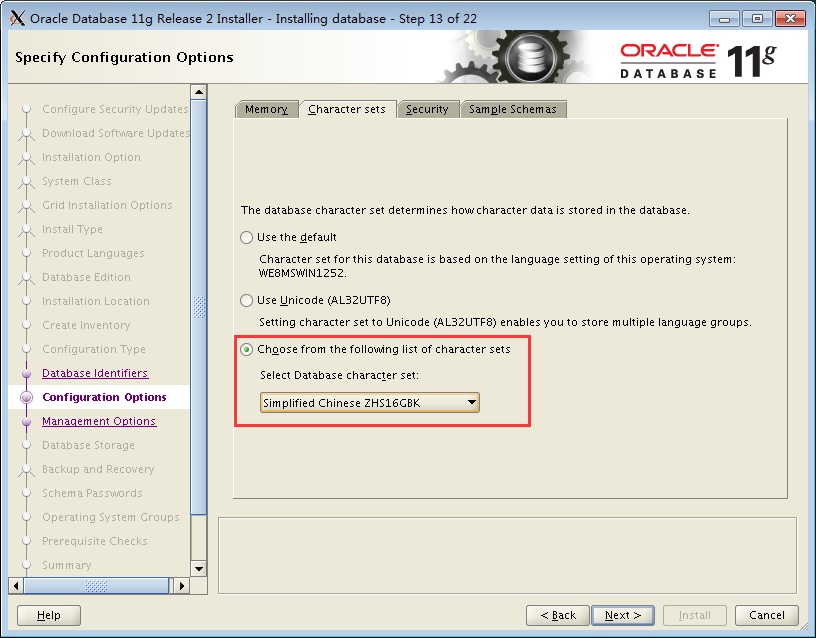
**(SERVICE\_NAME = orcl)**

**)**

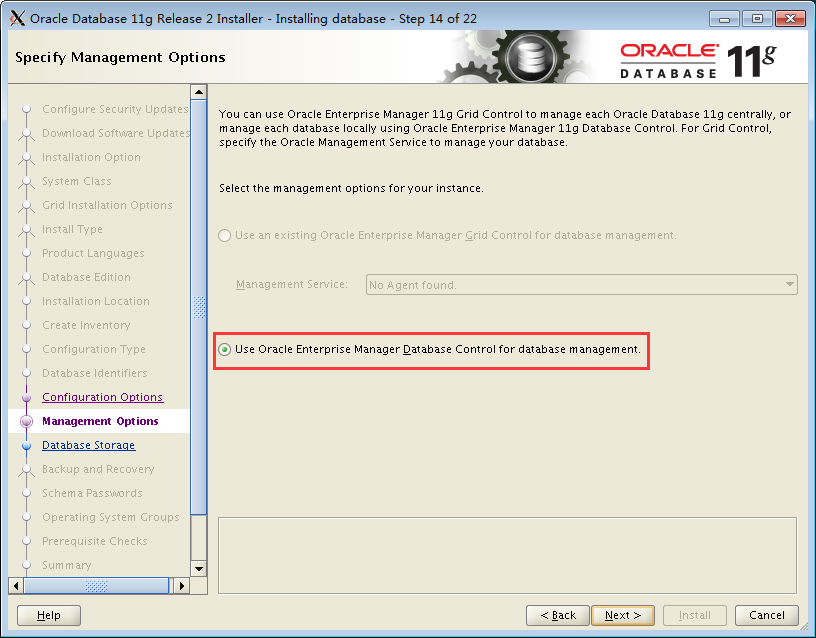
**)**

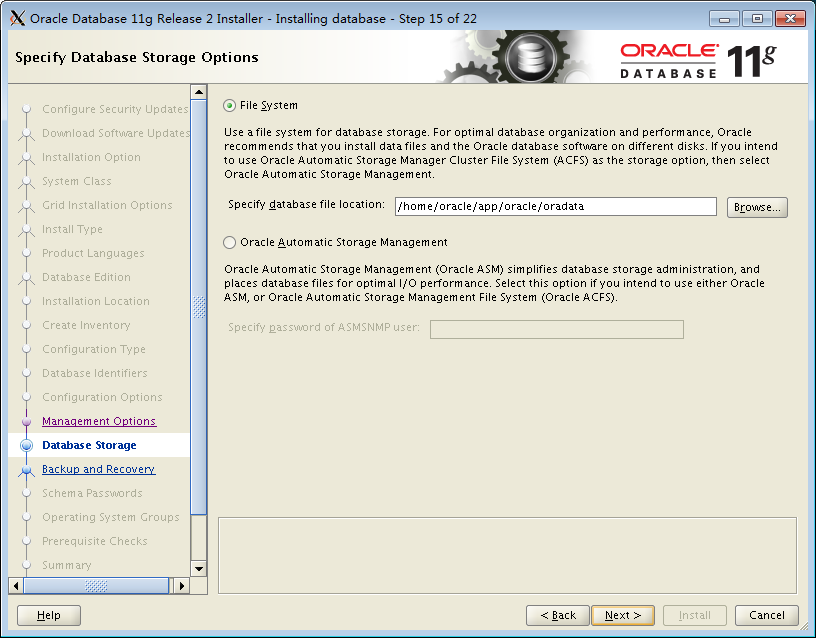


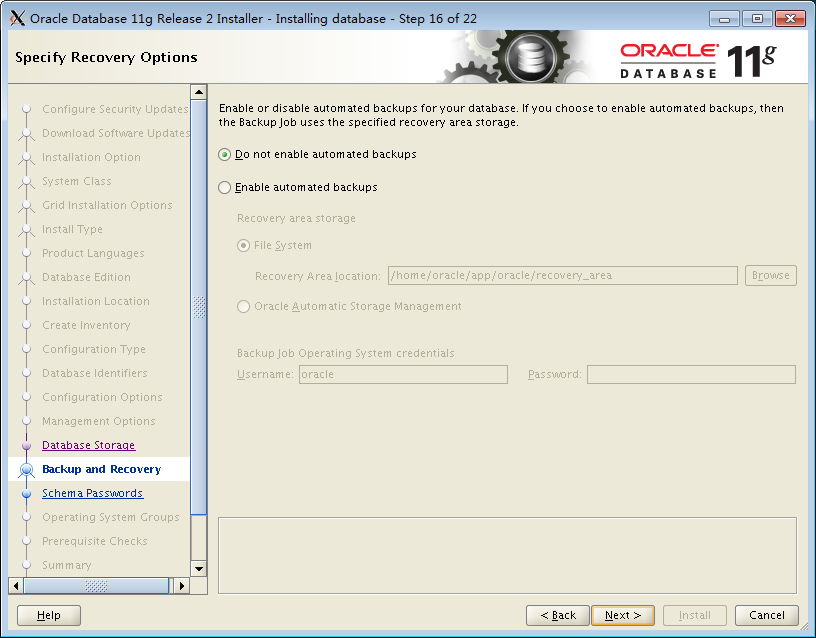
**分配数据库内存MEMORY\_TARGET**

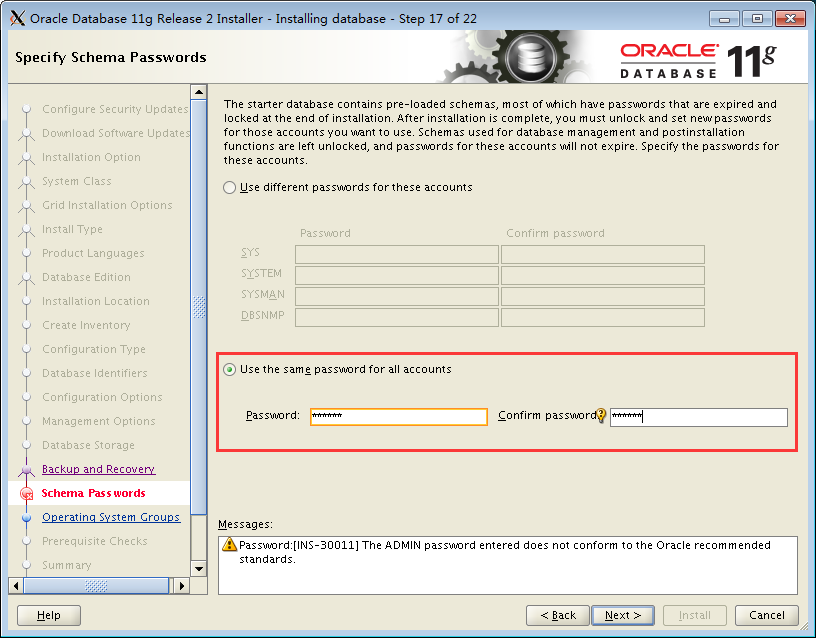


**注：此处是oracle字符集的设置，现在都统一使用ZHS16GBK的字符集**

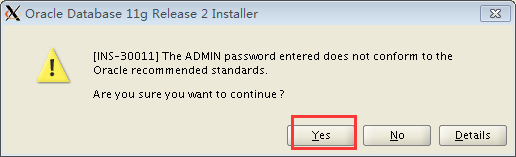




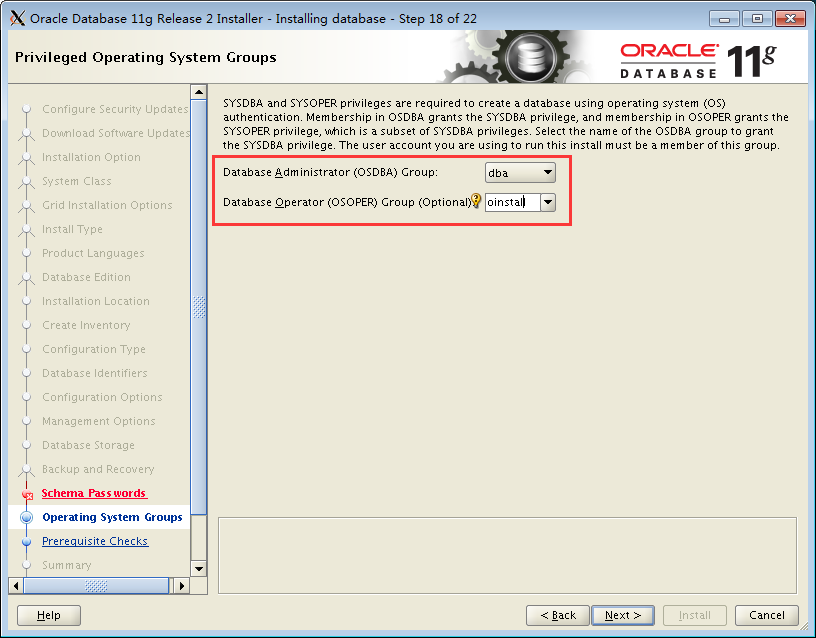


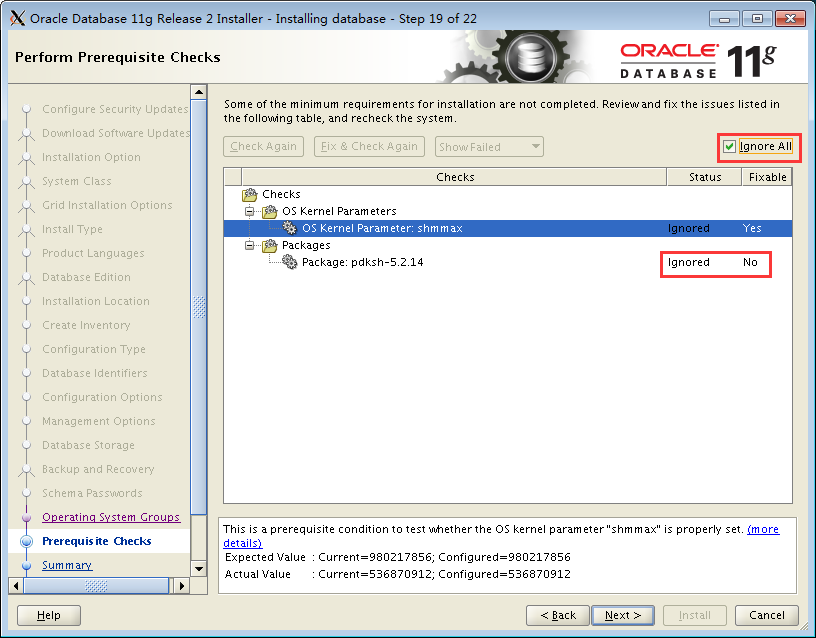


**注：此处可以简单点，方便记忆，选择下面的选项，将所有的用户都配置成同一个密码**

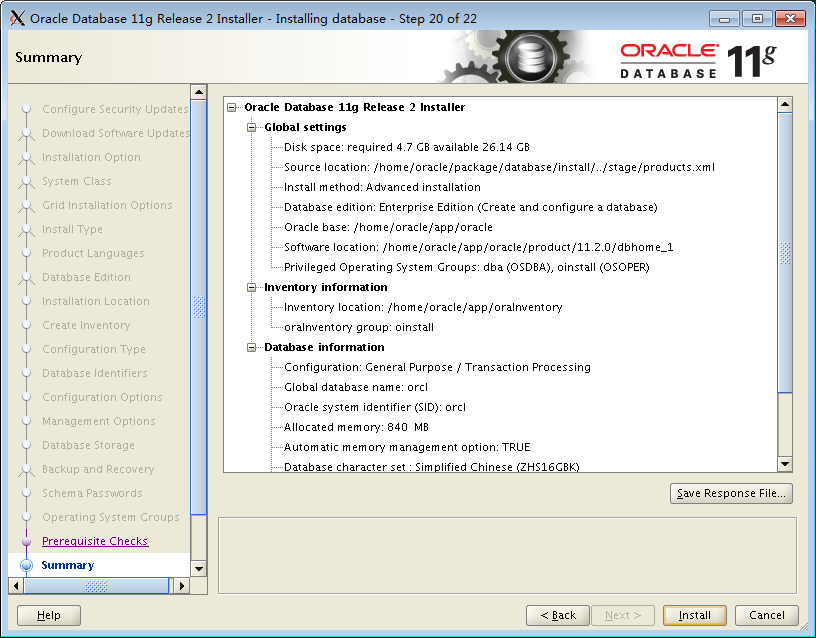


**注：会提示密码不符合标准，没有关系，点击Yes,继续安装**



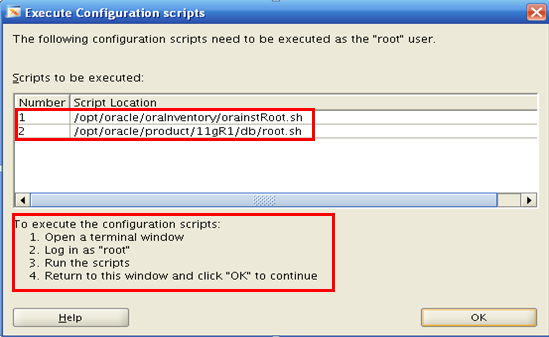


**注：此处就是前面说的，检测数据环境的问题，如果缺少上面，这里给出提示，然后给出提示进行相应的更改即可；如果有的包找不到，如pdksh，则勾选右上角的ignore**



**注：点击install，开始进行安装**

**安装完成后，出现以下界面，需要执行两个命令**

****

**按照上面的提示，打开终端，以root用户登录，运行以下两个sh文件**

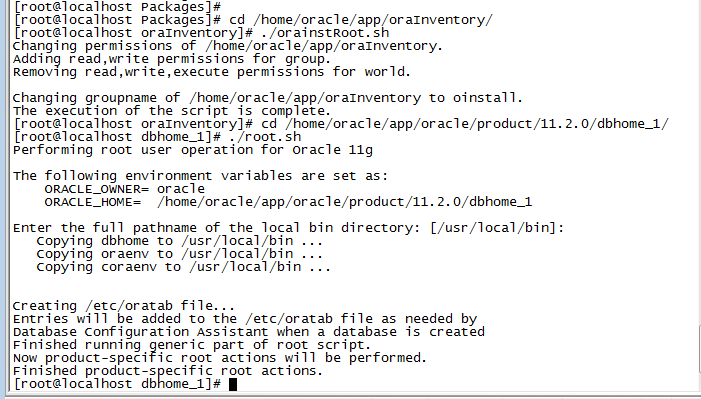
**过程中出现中间步骤，直接回车即可**

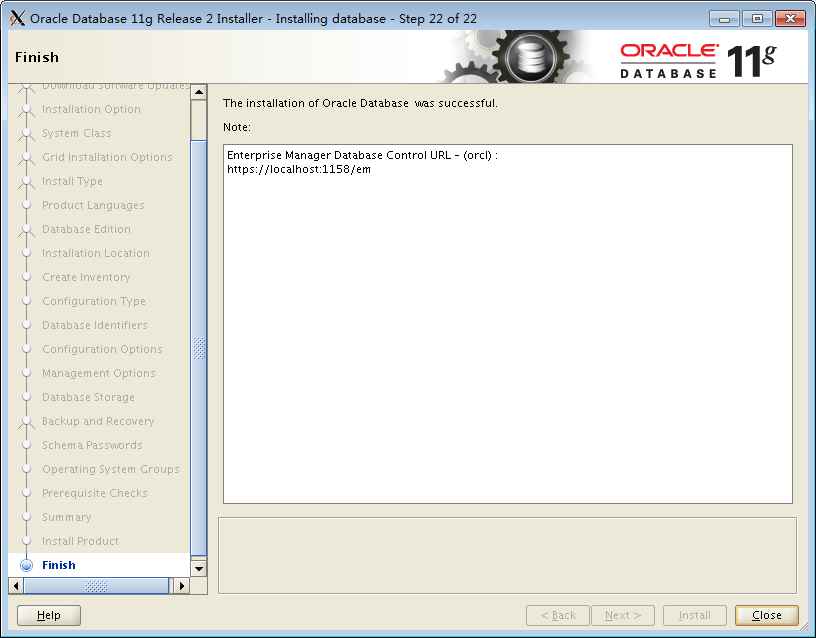
[root@localhost ~]# cd /home/oracle/app/oraInventory

[root@localhost oraInventory]# ./orainstRoot.sh

[root@localhost oraInventory]# cd /home/oracle/app/oracle/product/11.2.0/dbhome\_1

[root@localhost dbhome\_1]# ./root.sh





**完成结束后，数据库的服务，以及监听都是启动的**

至此，数据库的安装已经结束

# 配置环境变量

oracle安装完毕后需配置环境变量。

[oracle@localhost ~]$ cd ~

[oracle@localhost ~]$ vi .bash\_profile

根据实际安装路径配置环境变量，eg：

ORACLE\_BASE=/home/oracle/app/oracle

ORACLE\_HOME=/home/oracle/app/oracle/product/11.2.0/dbhome\_1

ORACLE\_SID=orcl

PATH=$ORACLE\_HOME/bin:$PATH

export ORACLE\_BASE

export ORACLE\_HOME

export ORACLE\_SID

export PATH

保存并退出：

:wq

使环境变量生效：

[oracle@localhost ~]$ source .profile

# 开启数据库的相关命令介绍

1. 开启oracle服务

连接sql：sqlplus /nolog

连接dba：conn /as sysdba

启动：startup

关闭：shutdown immediate

1. 开启监听

Oracle用户下，开启监听：lsnrctl start

关闭监听：lsnrctl stop

查看监听状态：lsnrctl status

# 数据库初始化

## 连接数据库

方法一：命令行下连接并使用sys用户登录：

先登录oracle用户，然后执行下列命令连接到sysdba用户

sqlplus /nolog

conn /as sysdba

（参考文档http://blog.sina.com.cn/s/blog\_9b0604b40101kk48.html)

方法二：使用plsql工具连接并使用sys用户登录：

直接使用plsql连接，登录到sys用户（这种方式需要sys用户的密码）

（注：oracle数据库linux和windows版本都可以通过plsql和命令行方式连接，操作都是一样的。）

## 创建表空间



执行顺序按照脚本中的顺序

脚本中的路径替换为实际路径，（建议统一路径为“/home/oracle/OracleData....”）

脚本可以在plsql用sys账户登录后直接执行脚本；也可以在命令行连接后逐条执行。

## 创建用户



脚本可以在plsql用sys账户登录后直接执行脚本；也可以在命令行连接后逐条执行。

## 导入数据

1. 使用SecureFXPortable工具将dump文件放到$ORACLE\_BASE 对应实例名的dpdump目录下，eg：$ORACLE\_BASE/admin/orcl/dpdump
2. 进到该目录

[oracle@localhost ~]$ cd $ORACLE\_BASE/admin/orcl/dpdump

1. 执行导入数据命令

(直接在命令行执行，先导md库，再导trd库)

[oracle@localhost dpdump]$ impdp xir\_md\_j/xpar@orcl dumpfile=XIR\_MD\_J\_20160426.DMP

[oracle@localhost dpdump]$ impdp trd\_cams/xpar@orcl dumpfile=TRD\_CAMS\_20160426.DMP

1. 导入数据命令说明

impdp 用户/密码@实例名 dumpfile=DMP文件

## 导出数据（初始化时不需要）

用于数据库备份、数据库迁移等。

(直接在命令行执行)

先md，后trd

expdp xir\_md\_j/xpar@xams dumpfile=XIR\_MD\_J\_20160426.DMP

expdp trd\_cams/xpar@xams dumpfile=TRD\_CAMS\_20160426.DMP

命令说明：

expdp 用户/密码@实例名 dumpfile=DMP文件

（导出文件默认放在$ORACLE\_BASE/admin/orcl/dpdump）

注：linux环境下执行导出命令如果报错（如ORA-39001,ORA-39000等），需先创建Directory对象，方法如下：

执行：

sqlplus /nolog

conn /as sysdba

进入sql命令行，

创建一个Directory对象，并授予用户读写权限。

SQL> create directory dump\_file as '/home/oracle/backup';（指定保存DMP文件的路径）

Directory created.

SQL> grant read,write on directory dump\_file to xir\_md\_j;（为导出数据的用户赋予权限）

Grant succeeded.

SQL> grant read,write on directory dump\_file to trd\_cams;（为导出数据的用户赋予权限）

Grant succeeded.

如上，生成的DMP文件将会放在/home/oracle/backup目录下。

# 重新导入数据

如果需要将原有数据库全部清除然后重新导入一份，可通过删除用户，然后重建用户再导入新的数据库。

1. 断开要删除用户的所有连接

--查看所有用户

select username from dba\_users;

--断开用户的所有相关进程（如：TRD\_CAMS用户）

alter user TRD\_CAMS account lock;

SELECT \* FROM V$SESSION WHERE USERNAME='TRD\_CAMS';

alter system kill session '70,123';（’SID,SERIAL#’）

（注：也可直接shutdown然后重启数据库）

1. 连接数据库

参考9.1节连接数据库

1. 删除用户

删除用户会导致数据丢失，执行前记得备份哦~，备份数据库参考9.5节导出数据。

删除用户也是先md，后trd。

DROP USER XIR\_MD\_J CASCADE;

DROP USER TRD\_CAMS CASCADE;

1. 重新创建用户

参考9.3节创建用户

1. 重新导入数据

参考9.4节导入数据

# 遇到的问题及解决办法

1. 数据库查询/前台界面乱码问题

**原因：**用plsql执行查询或升级脚本，本地未配置环境变量

**解决办法：**为plsql所在的本地配置环境变量：nls\_lang=simplified chinese\_china.ZHS16GBK

1. 启动数据库问题

开启oracle服务，startup报错：

ORA-01078: failure in processing system parameters

LRM-00109: could not open parameter file '$ORACLE\_HOME/dbs/initORCL.ora'

**解决办法：**

复制$ORACLE\_BASE/admin/orcl/pfile下的init.ora.731201291548到$ORACLE\_HOME/dbs下，并更改成需要的名称initORCL.ora

1. 启动数据库报错

ORA-00845: MEMORY\_TARGET not supported on this system

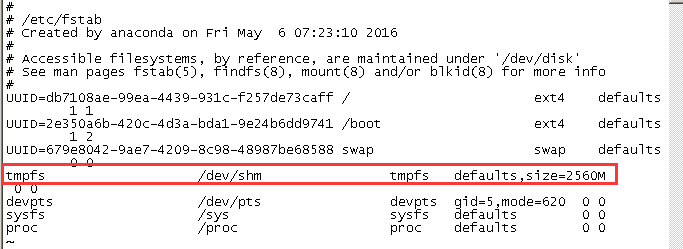
**原因：**是因为/dev/shm的可用空间（非shm的总大小）小于了参数文件中设置的MEMORY\_TARGET值。

**解决办法：**增加/dev/shm，使其大于MEMORY\_TARGET值

需root用户

[root@localhost ~]#vi /etc/fstab

修改tmpfs对应的/dev/shm大小，如下：



保存：

:wq

重新挂载/dev/shm

[root@localhost ~]#mount -o remount /dev/shm

可用df -h 命令查看修改结果

[root@localhost ~]#df -h

然后回到oracle用户重新启动。

1. SWAP空间不足的问题

内存大小要求：

内存RAM最小为1G

SWAP大小要求：

1G<RAM<2G时，SWAP为RAM大小的1.5倍

2G<RAM<16G时，SWAP等于RAM大小

RAM>=16G时，SWAP为16G

SWAP空间不够的解决方法

例：增加1G SWAP空间

添加交换文件

[root@localhost ~]#dd if=/dev/zero of=/home/oracle/swap bs=1024 count=1024

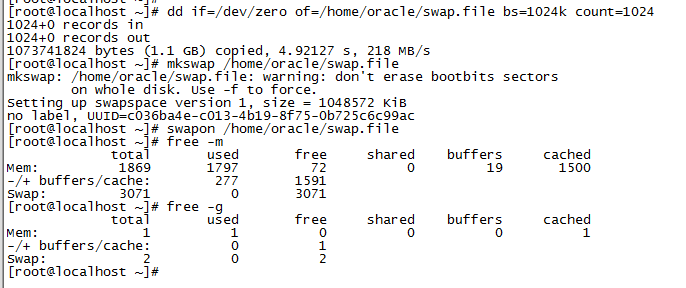
处事换交换空间

[root@localhost ~]#mkswap /home/oracle/swap.file

启动新增加的1G交换空间

[root@localhost ~]#swapon /home/oracle/swap.file

[root@localhost ~]#free -m



修改/etc/fstab,使新加的1G交换空间在系统重新启动后自动生效

[root@localhost ~]#echo "/home/oracle/swap.file swap swap defaults 0 0" >>/etc/fstab

（参考文档：<http://www.linuxidc.com/Linux/2015-12/126602.htm>）

1. ORA-01102: cannot mount database in EXCLUSIVE mode

参考文档：<http://www.linuxidc.com/Linux/2014-02/96744.htm>

参考以上文档不能解决，可以kill掉所有oracle进程然后重启数据库

1. ORA-32001 write to SPFILE requested but no SPFILE is in use

参考文档：<http://www.techpaste.com/2012/12/ora-32001-write-spfile-requested-spfile-oracle-database>

7. 安装ORACLE11.2.0.4报错

安装时到70%左右会出现一个错误:Error in invoking target 'agent nmhs' of makefile '/u01/app/oracle/product/1

解决办法：保留安装过程，另外开启一个终端窗口，将ins\_emagent.mk文件中的 (MK\_EMAGENT\_NMECTL)更改为$(MK\_EMAGENT\_NMECTL) -lnnz11，然后在安装过程中点击Retry即可。

vi $ORACLE\_HOME/sysman/lib/ins\_emagent.mk

Search for the line

$(MK\_EMAGENT\_NMECTL)

Change it to:

$(MK\_EMAGENT\_NMECTL) -lnnz11

8. ORA-03113：通信通道的文件结尾

参考文档：

<https://www.cnblogs.com/wxfistudy/p/6638514.html>

9. ORA-00257归档日志已满

参考文档：

https://blog.csdn.net/u013519551/article/details/70256793

10. ORACLE查看被锁的表和解锁

**--以下几个为相关表**  
SELECT \* FROM v$lock;  
SELECT \* FROM v$sqlarea;  
SELECT \* FROM v$session;  
SELECT \* FROM v$process ;  
SELECT \* FROM v$locked\_object;  
SELECT \* FROM all\_objects;  
SELECT \* FROM v$session\_wait;

--查看被锁的表   
select b.owner,b.object\_name,a.session\_id,a.locked\_mode from v$locked\_object a,dba\_objects b where b.object\_id = a.object\_id;

--查看那个用户那个进程照成死锁  
select b.username,b.sid,b.serial#,logon\_time from v$locked\_object a,v$session b where a.session\_id = b.sid order by b.logon\_time;

--查看连接的进程   
SELECT sid, serial#, username, osuser FROM v$session;

--查出锁定表的sid, serial#,os\_user\_name, machine\_name, terminal，锁的type,mode  
SELECT s.sid, s.serial#, s.username, s.schemaname, s.osuser, s.process, s.machine,  
s.terminal, s.logon\_time, l.type  
FROM v$session s, v$lock l  
WHERE s.sid = l.sid  
AND s.username IS NOT NULL  
ORDER BY sid;

这个语句将查找到数据库中所有的DML语句产生的锁，还可以发现，  
任何DML语句其实产生了两个锁，一个是表锁，一个是行锁。

--杀掉进程 sid,serial#  
alter system kill session'210,11562';

11. ORA-01034和ORA-27101

参考文档：

https://blog.csdn.net/u013519551/article/details/53485342