GUIDING INSTALL
ORACLE & OCI8 & PDO
On Server CentOS 6.4

Version 0.0.1

1. Install Oracle

Installing Oracle Database 11g Release 2

Step 1: Install Oracle Dependencies

We use “oracle-rdbms-server-11gR2-preinstall” package offered by “Oracle Public Yum” repository. The Oracle
public yum repository provides a free and easiest way to install all the latest Oracle Linux dependencies automatically.
To setup yum repository, follow the instructions provided below.
Use “wget” command to Download appropriate yum configuration file under /etc/yum.repos.d/ directory as root user.

RHEL/CentOs 6.x

```
# cd /etc/yum.repos.d
# wget https://public-yum.oracle.com/public-yum-ol6.repo
```

Now perform the following “yum” command to install all the necessary prerequisites automatically.

```
[root@oracle]# yum install oracle-rdbms-server-11gR2-preinstall
```

While importing GPG key, you might get “GPG key retrieval failed” error as shown below. Here, you need to import proper GPG key for your OS release. Retrieving key from
file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
GPG key retrieval failed: [Errno 14] Could not open/read
file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
Download and verify the appropriate Oracle Linux GPG Key that best matches your RHEL/CentOS compatible OS release.

RHEL/CentOs 6.x

```
# wget https://public-yum.oracle.com/RPM-GPG-KEY-oracle-ol6 -O
/etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
```

Step 2: Setting Hostname

Open the “/etc/sysconfig/network” file and modify the HOSTNAME to match your FQDN (Fully Qualified Domain Name) host name.

```
[root@oracle]# vi /etc/sysconfig/network
HOSTNAME=oracle.tecmint.com
```

Open “/etc/hosts” file and add fully qualified hostname for the server.

```
[root@oracle]# vi /etc/hosts
192.168.246.128
3 of 49
oracle.tecmint.com
oracle
```

Now you need to restart networking on the server to make sure that changes will be persistent on reboot.

```
[root@oracle]# /etc/init.d/network restart
```

Step 3: Oracle User Settings

Add group and user oracle

```
# groupadd oinstall
# groupadd dba
# useradd -d /opt/oracle -g oinstall -G dba -s /bin/bash oracle
# passwd oracle
Changing password for user oracle.
New UNIX password:
BAD PASSWORD: it is based on a dictionary word
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

Add the entry to file "/etc/security/limits.d/90-nproc.conf" as described below.

```
[root@oracle]# vi /etc/security/limits.d/90-nproc.conf
# Default limit for number of user's processes to prevent
# accidental fork bombs.
# See rhbz #432903 for reasoning.
*
soft
# To this
* - nproc 16384
nproc
1024
```

Set SELinux to "permissive" mode by editing the file "/etc/selinux/config".

```
[root@oracle]# vi /etc/selinux/config
SELINUX=permissive
```

Once you've made change, don't forger to restart the server to reflect new changes.

```
[root@oracle]# reboot
```

Login as Oracle user and open file ".bash_profile", which is available on oracle user's home directory, make an entries as described below. Make sure you set correct hostname to "**ORACLE_HOSTNAME=oracle.tecmint.com**".

```
[root@oracle]# su oracle
[oracle@oracle ~]$ vi .bash_profile
# Oracle Settings
TMP=/tmp; export TMP
TMPDIR=$TMP; export TMPDIR
ORACLE_HOSTNAME=oracle.tecmint.com; export ORACLE_HOSTNAME
ORACLE_UNQNAME=DB11G; export ORACLE_UNQNAME
ORACLE_BASE=/u01/app/oracle; export ORACLE_BASE
ORACLE_HOME=$ORACLE_BASE/product/11.2.0/dbhome_1; export ORACLE_HOME
```

```
ORACLE_SID=DB11G; export ORACLE_SID
PATH=/usr/sbin:$PATH; export PATH
PATH=$ORACLE_HOME/bin:$PATH; export PATH
LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib; export LD_LIBRARY_PATH
CLASSPATH=$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib; export CLASSPATH
export PATH
```

Switch to root user and issue the following command to allow Oracle user to access X Server.

```
[root@oracle]# xhost +
```

Create the directories and set the appropriate permissions in which the Oracle software will be installed.

```
[root@oracle]# mkdir -p /u01/app/oracle/product/11.2.0/dbhome_1
[root@oracle]# chown -R oracle:oinstall /u01
[root@oracle]# chmod -R 775 /u01
```

Step 4: Downloading Oracle Software

Sign-up and Download the Oracle software using the following link. Download Oracle 11g Release 2

The Oracle package contains 2 zip files which you must first accept the license agreement before downloading. I've given the files names for you reference, please download these files for your system architecture somewhere under "**/home/oracle/**".

For Linux x86 Systems

http://download.oracle.com/otn/linux/oracle11g/R2/linux_11gR2_database_1of2.zip
http://download.oracle.com/otn/linux/oracle11g/R2/linux_11gR2_database_2of2.zip

For Linux x86-64 Systems

http://download.oracle.com/otn/linux/oracle11g/R2/linux.x64_11gR2_database_1of2.zip
http://download.oracle.com/otn/linux/oracle11g/R2/linux.x64_11gR2_database_2of2.zip

In my case, I always do some steps below to make sure all configuration are ready before install Oracle. All the steps as below:

Step 5: The second way to preparation

Install a X server

```
yum groupinstall "X Window System"
```

Or, for a remote install on a server that doesn't need a graphical interface, install the following packages and dependencies to do X11 forwarding:

```
yum install xorg-x11-xauth.x86_64 xorg-x11-apps.x86_64
```

Change the timezone to GMT

```
# rm /etc/localtime
# ln -s /usr/share/zoneinfo/GMT /etc/localtime
```

Install some additional packages (yes, oracle installer requires GCC.... :/)

```
# yum install libXp libXtst binutils compat-db compat-libstdc++-33 glibc glibc-devel
glibc-headers gcc gcc-c++ libstdc++ cpp make libaio ksh elfutils-libelf sysstat libai
```

On CentOS 6 64 bits, I also found the glibci686 to be necessary, otherwise /lib/ld-linux.so.2 is missing:

```
yum install glibc-2.12-1.7.el6_0.5.i686
```

Add limits to /etc/security/limits.conf

```
# grep oracle /etc/security/limits.conf
# specific to oracle
oracle   soft     nproc   2047
oracle   hard     nproc   16384
oracle   soft     nofile  1024
oracle   hard     nofile  65536
```

For limits to be applied on login in /etc/pam.d/login

```
# grep limit /etc/pam.d/login
session required /lib64/security/pam_limits.so
```

Modify the SYSCTL parameters in /etc/sysctl.conf

```
# oracle sysctl parameters ------------------
# max number of file descriptors
fs.file-max = 65535
# maximum size in bytes of a single shared memory segment
# that a Linux process can allocate in its virtual address space.
# with 64GB of RAM, we set the max segment size at 60GB, thus the
# SGA cannot be larger than 60GB for one database instance
kernel.shmmax = 64424509440
# total amount of shared memory pages that can be used system wide.
# Hence, SHMALL should always be at least ceil(shmmax/PAGE_SIZE).
# with 60GB of RAM and a page size at 4KB
kernel.shmall = 15728640
# system wide maximum number of shared memory segments
kernel.shmmni = 4096

# control the number of semaphores on the system
# kernel.sem = semmsl semmns semopm semmni
# SEMMSL kernel parameter is used to control the maximum number
# of semaphores per semaphore set.
# SEMMSL setting should be 10 plus the largest PROCESSES
# parameter of any Oracle database on the system, here 2048
# SEMMNS parameter is the maximum number of semaphores that
# can be allocated (SEMMSL * SEMMNI) system wide
# SEMOPM kernel parameter is used to control the number of
# semaphore operations that can be performed per semop system call
# SEMMNI kernel parameter is used to control the maximum number
# of semaphore sets on the entire Linux system.
kernel.sem = 2048 262144 2048 128
```

```
# usable port range
net.ipv4.ip_local_port_range = 9000 65500
# default OS receive buffer size in bytes
net.core.rmem_default = 262144
# max OS receive buffer size in bytes
net.core.rmem_max = 4194304
# default OS transmit buffer size in bytes
net.core.wmem_default = 262144
# max OS transmit buffer size in bytes
net.core.wmem_max = 1048576
```

And maybe remove SELINUX

```
# vim /etc/sysconfig/selinux
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
# enforcing - SELinux security policy is enforced.
# permissive - SELinux prints warnings instead of enforcing.
# disabled - SELinux is fully disabled.
SELINUX=disabled
# SELINUXTYPE= type of policy in use. Possible values are:
# targeted - Only targeted network daemons are protected.
# strict - Full SELinux protection.
SELINUXTYPE=targeted
# SETLOCALDEFS= Check local definition changes
SETLOCALDEFS=0
```

Step 6: Oracle Installation

Now let's start Oracle installation. First of all need to switch as 'oracle' user to install database.

```
[oracle@oracle ~]$ su oracle
```

Extract compressed Oracle database source files to the same directory "/home/oracle/".

```
[oracle@oracle ~]$ unzip linux_11gR2_database_1of2.zip
[oracle@oracle ~]$ unzip linux_11gR2_database_2of2.zip
```

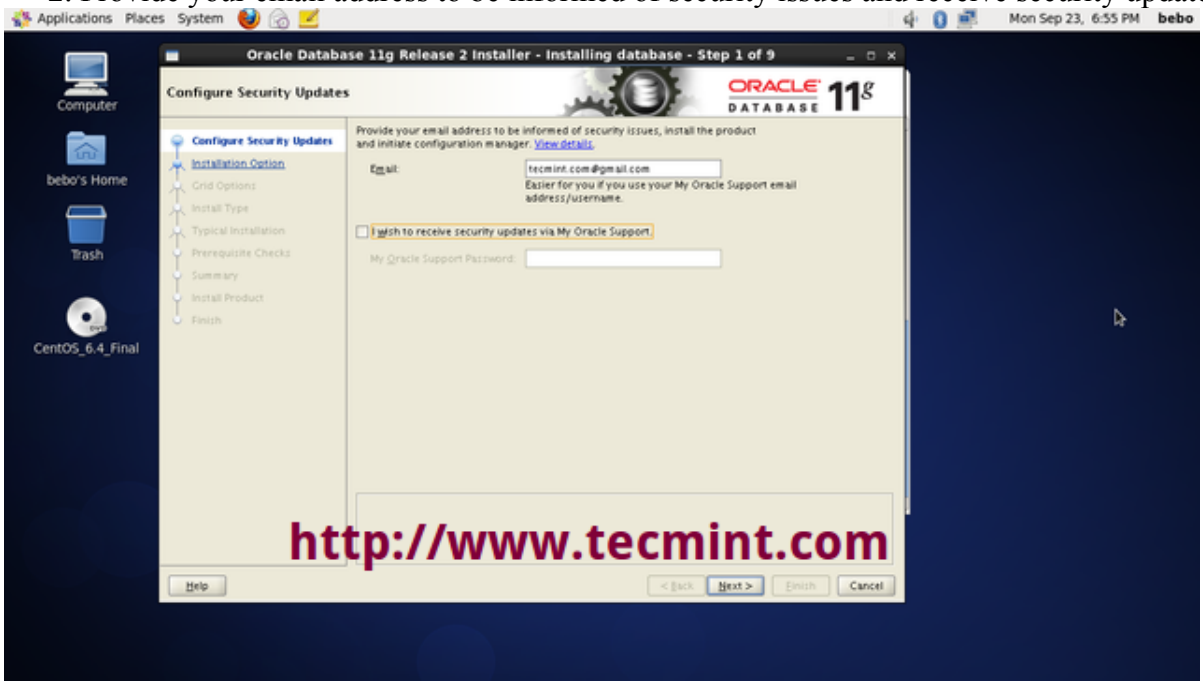
Post unzip source file, directory called database will be created, go to inside the directory and execute below script to start Oracle database installation process.

```
[oracle@oracle database]$ cd database
[oracle@oracle database]$ ./runInstaller
```

1. RunInstaller will call Oracle Universal Installer (OUI), wherein look and feel & steps are the same across all the operating system.

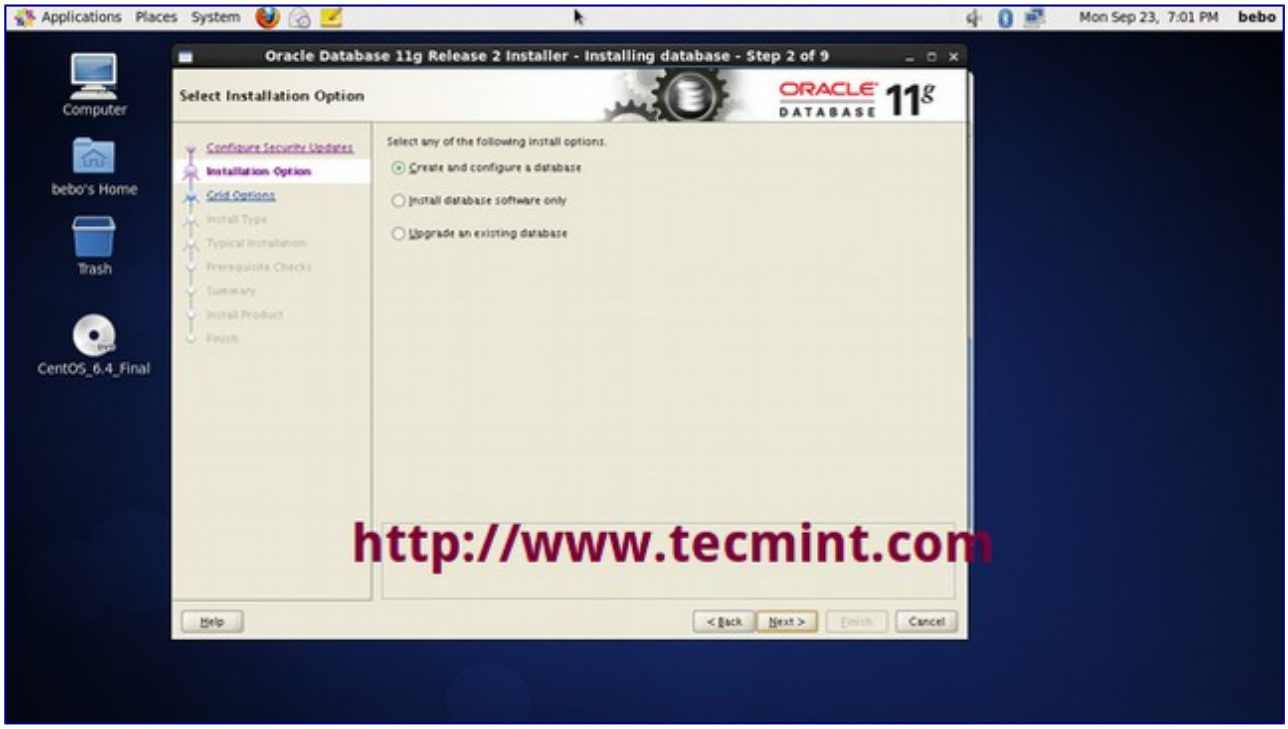
Oracle Universal Installer

2. Provide your email address to be informed of security issues and receive security updates.



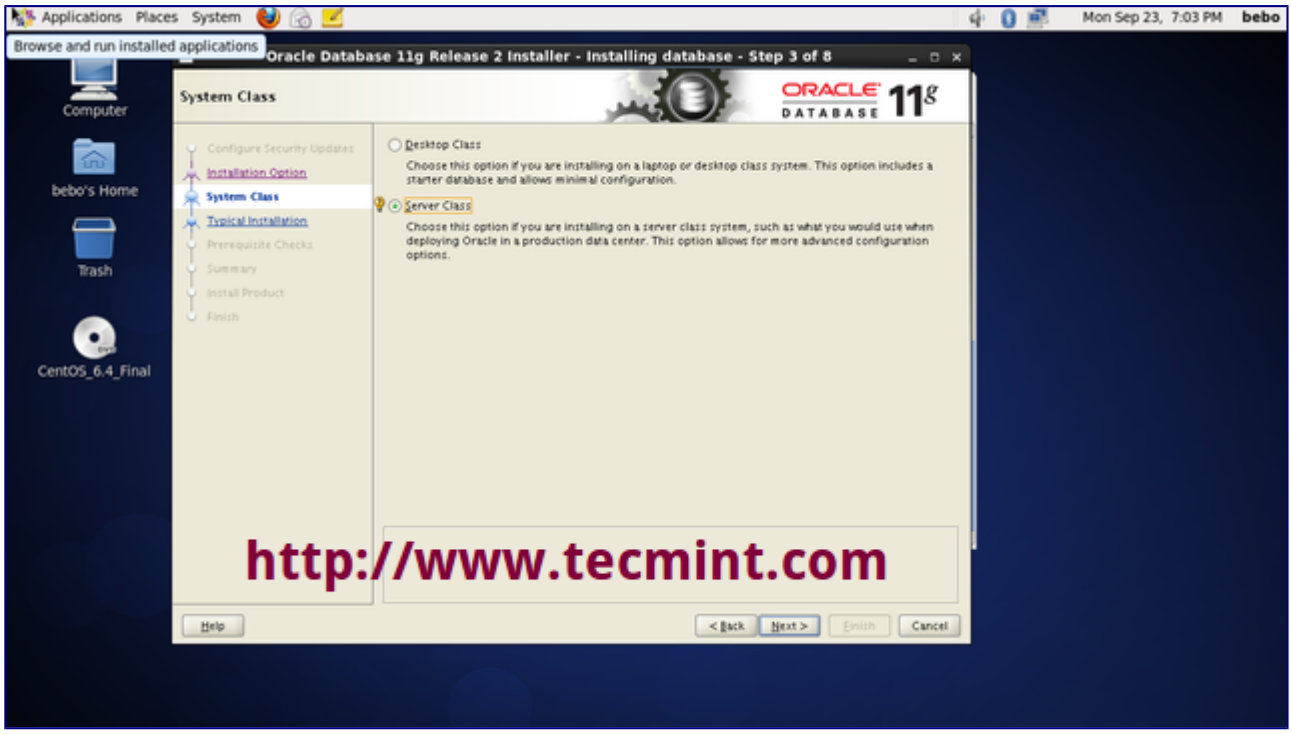
Configure Oracle Security Updates

3. Create and Configure a **Database**



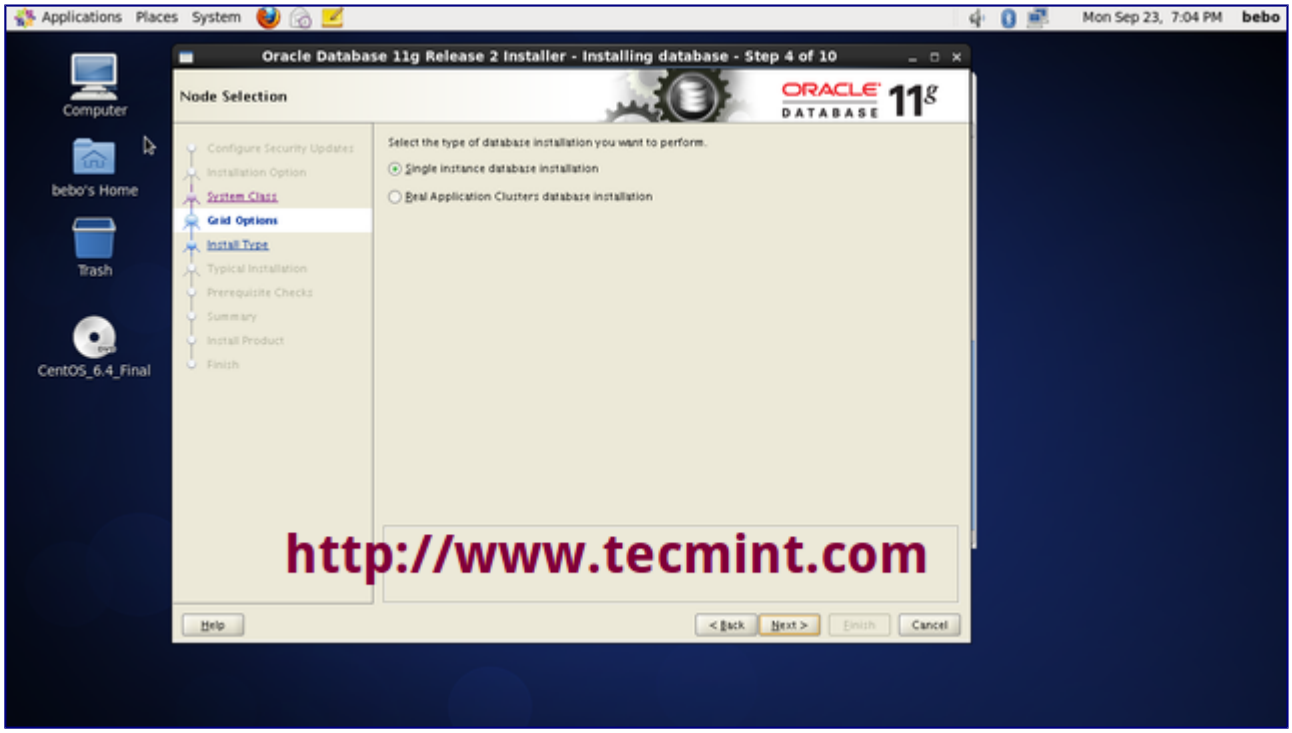
Create Oracle Database

4. Choose the system class, either **Desktop** or **Server**.



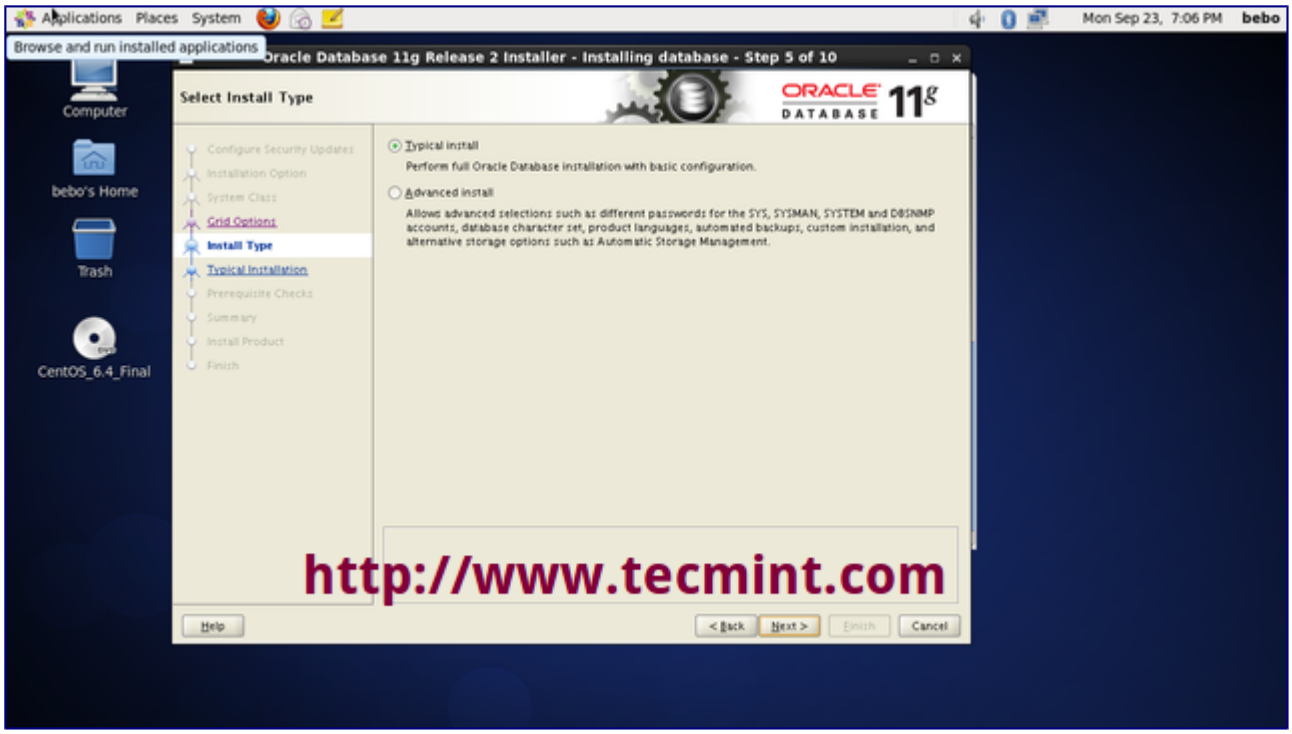
Select Oracle System Class

5. Select the type of **database installation** you want to perform.



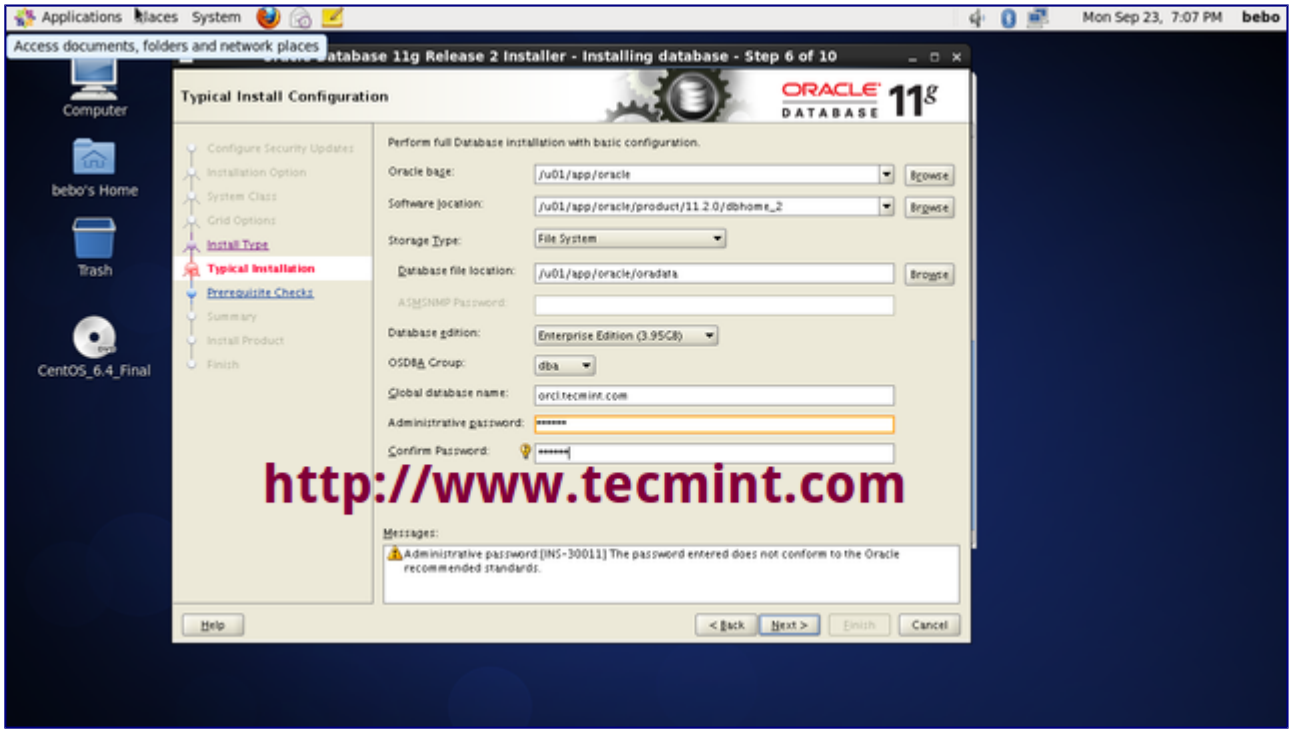
Select Database Installation Type

6. Select "**Typical install**" option to install full oracle installation with basic configuration.



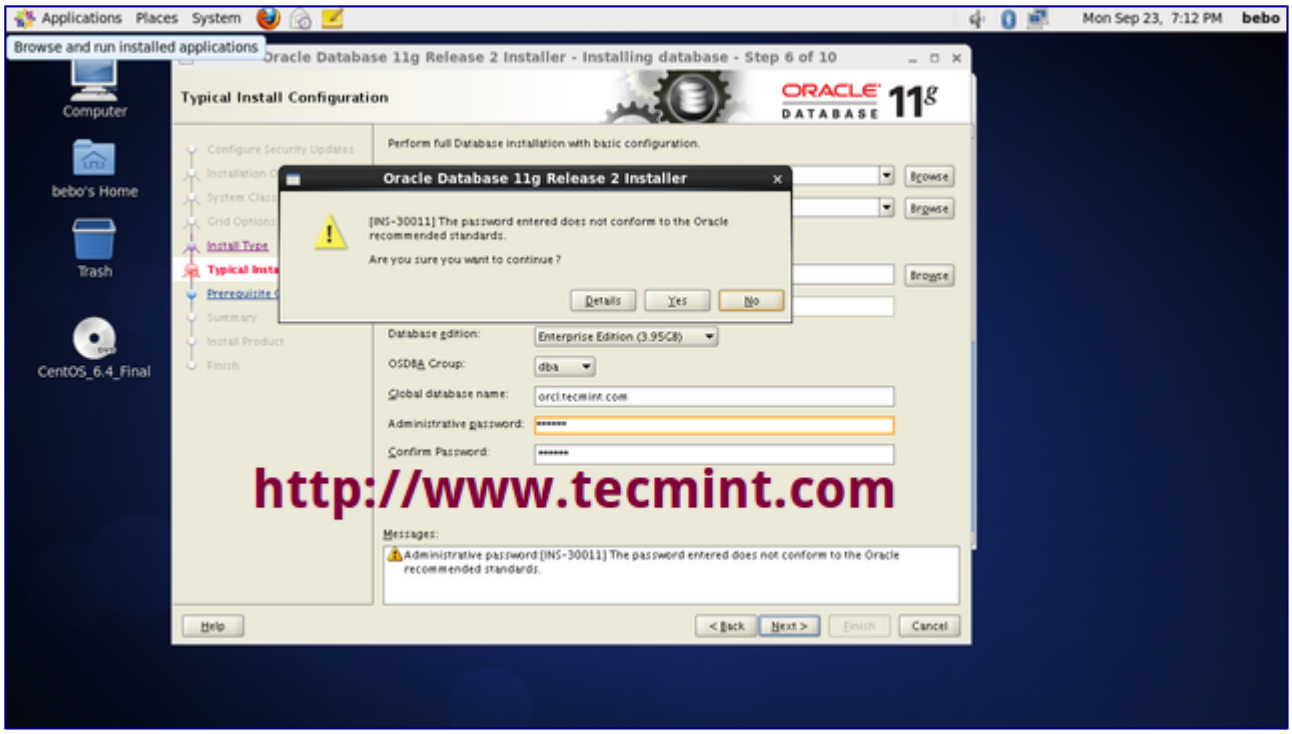
Select Typical Install

7. Set **Administrative password** and perform full Database installation with basic configuration.



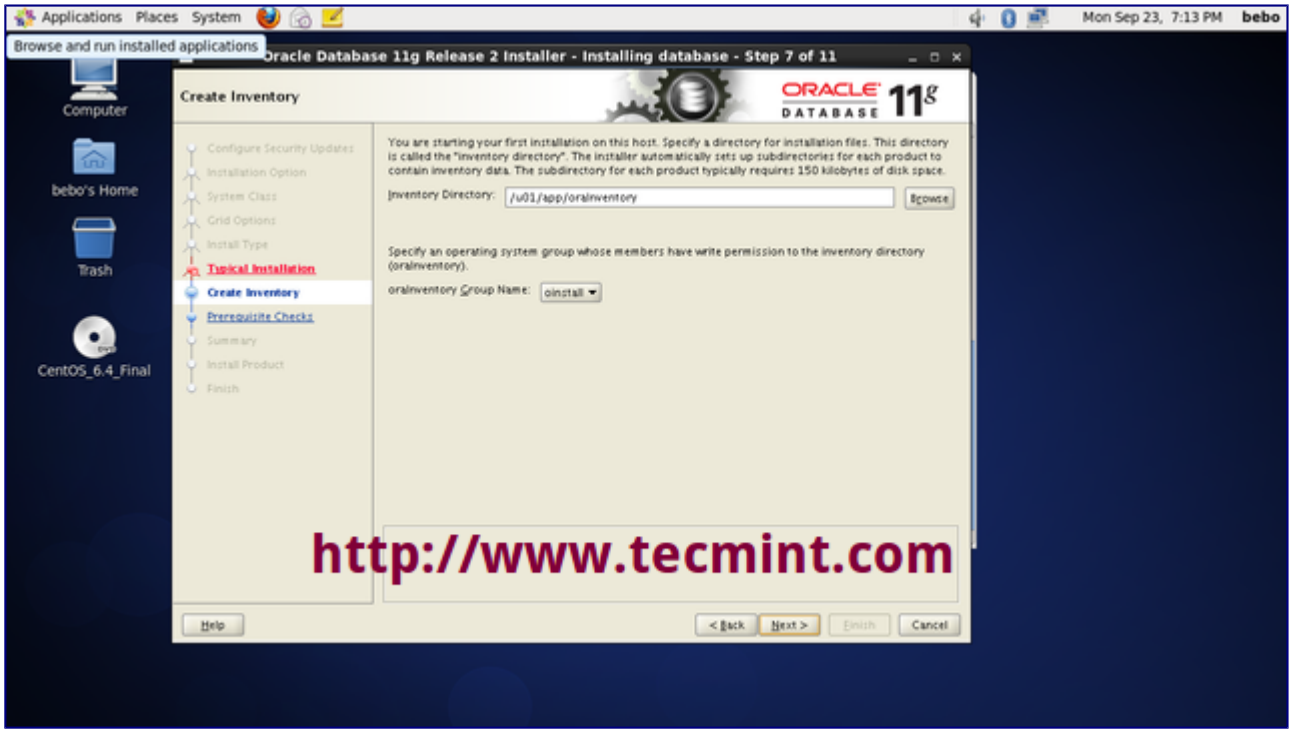
Set Oracle Administrative Password

8. Please click on "**Yes**" to continue with installation.



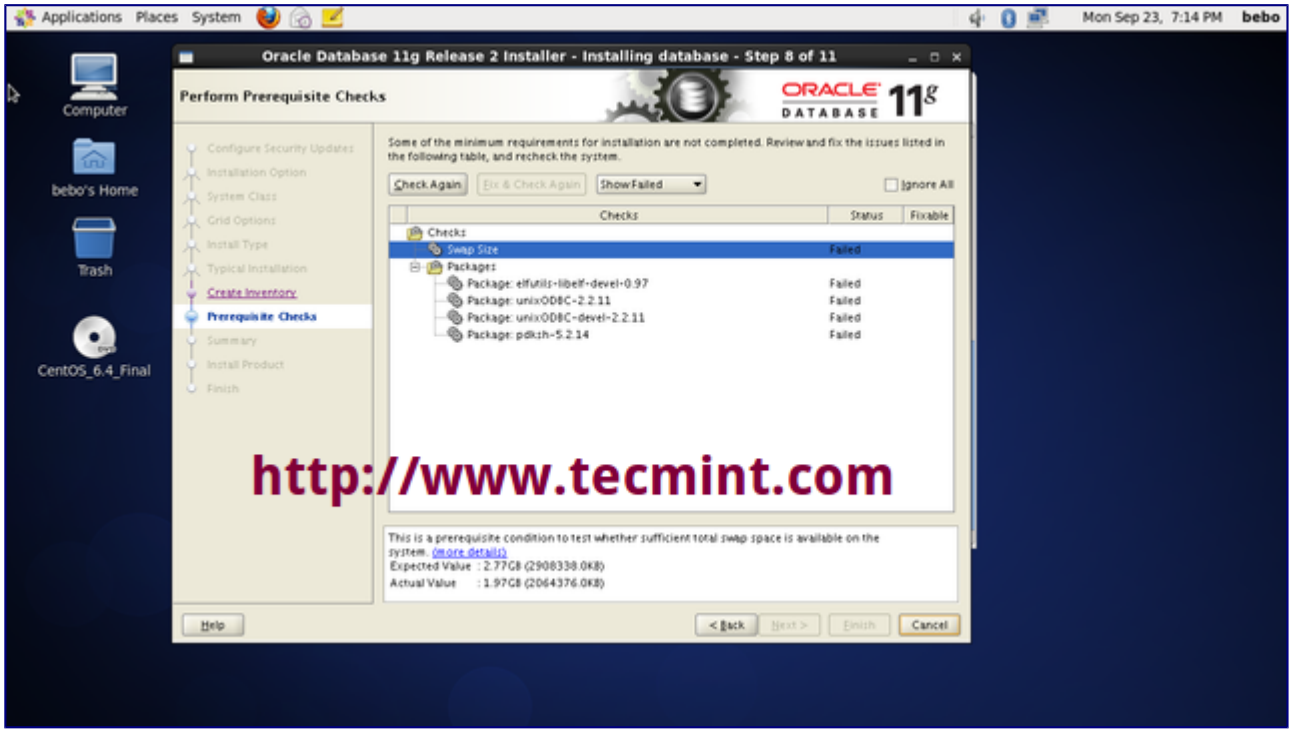
Typical Install Configuration Confirm

9. Create **Inventory**



Create Inventory

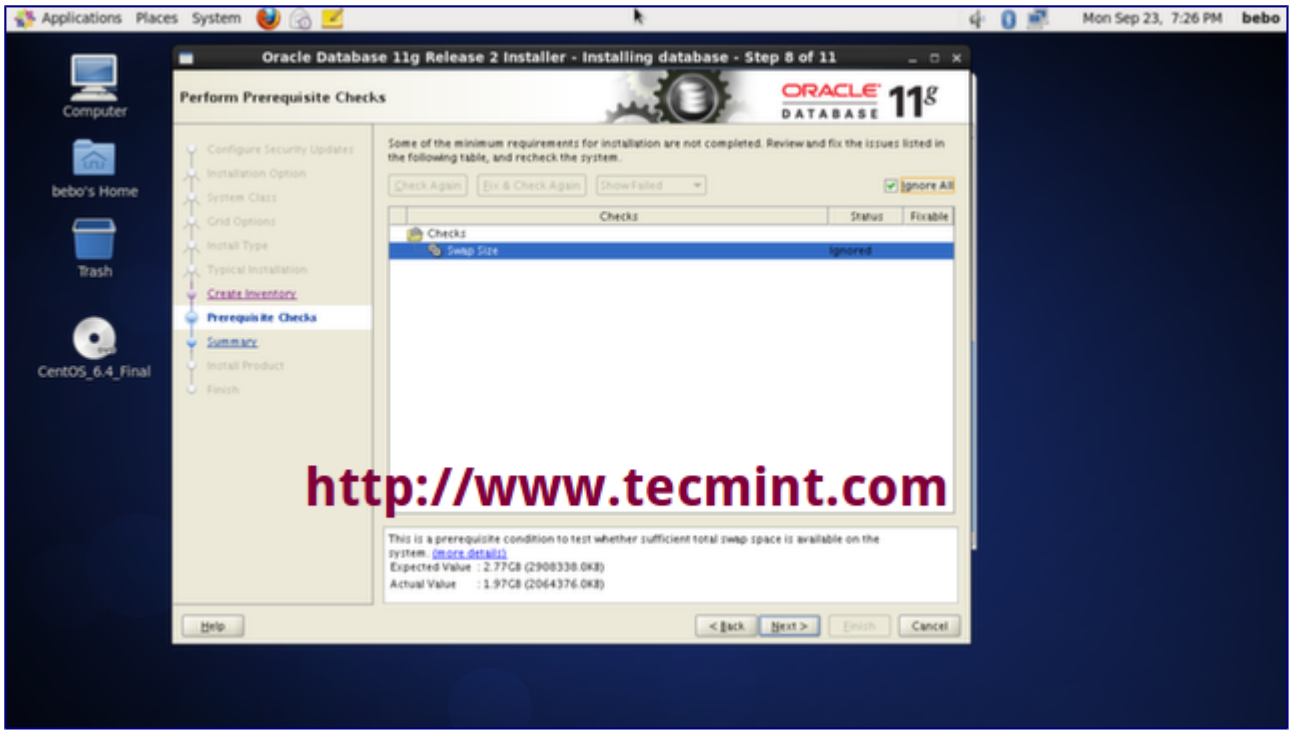
10. If you faced prerequisites warning during installation. Click on "**Fix & Check Again**". Oracle fixes Prerequisites by itself. This is the new feature of **Oracle Database 11g**.



Perform Prerequisite Checks

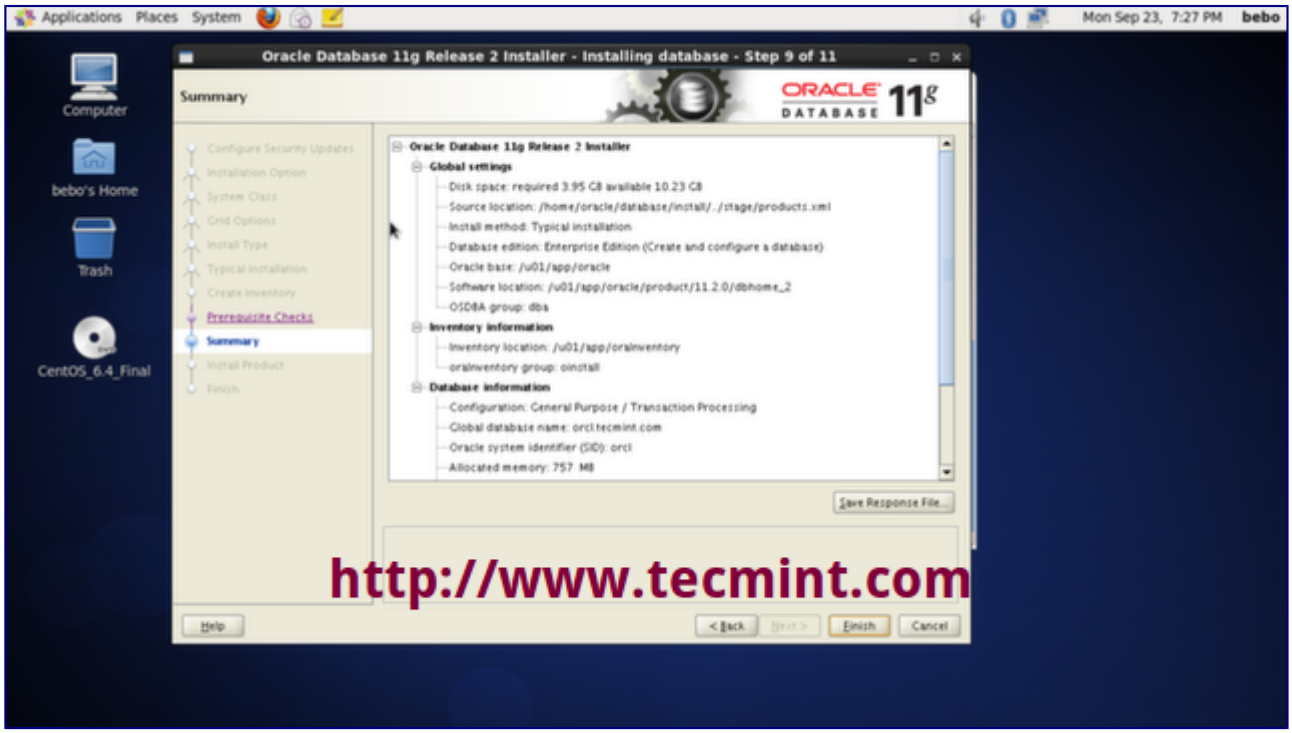
11. Performing Prerequisite checks: It's test whether sufficient total **SWAP** space is

available on the system.



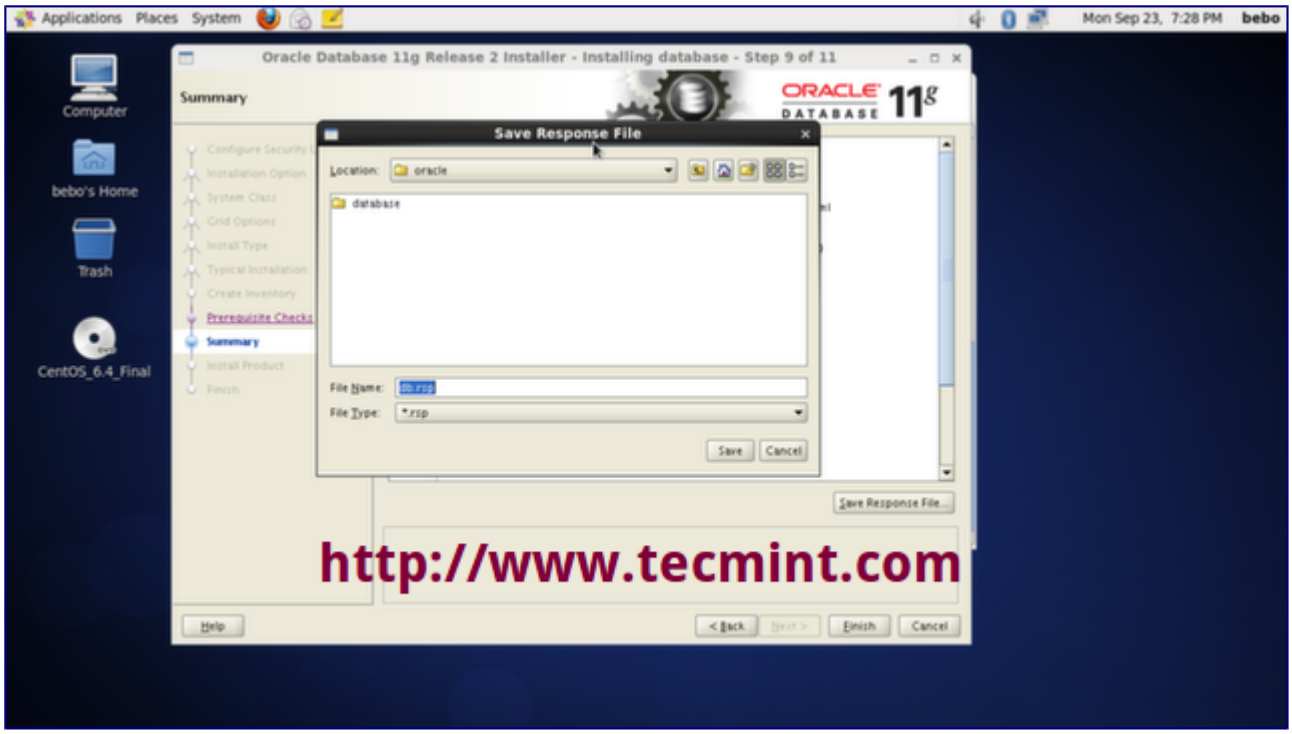
Performing Prerequisite SWAP Check

12. Installation Summary: Click on **Save Response File**. This file is useful for **Oracle Silent Mode Installation**



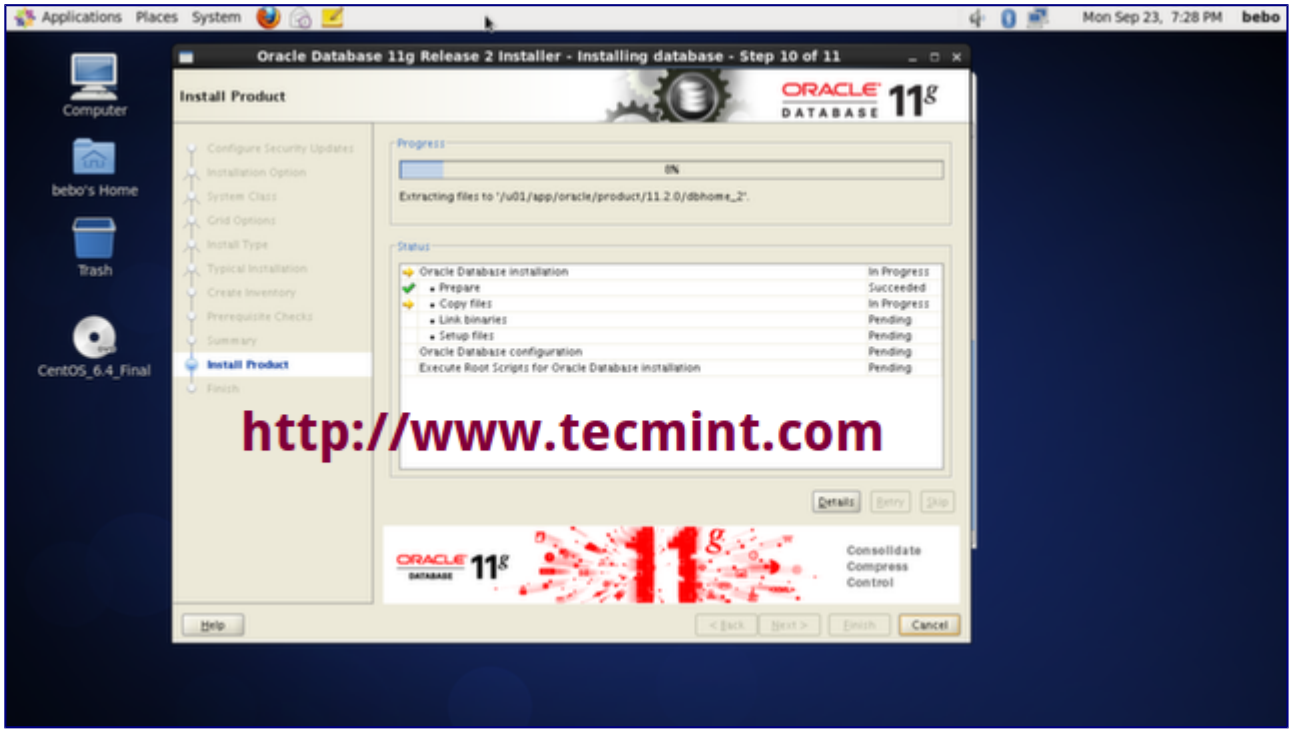
Oracle Installation Summary

13. Save Response File somewhere in your system.



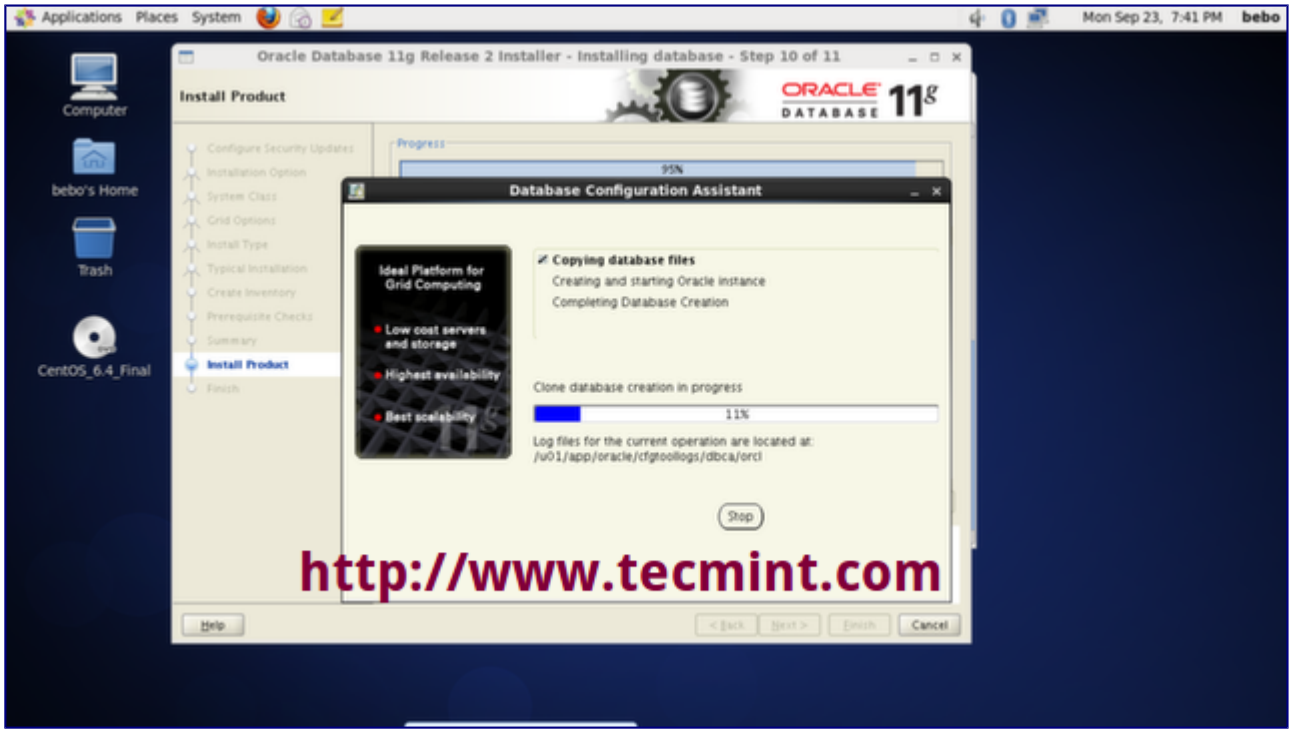
Save Response File

14. Product Installation Progress



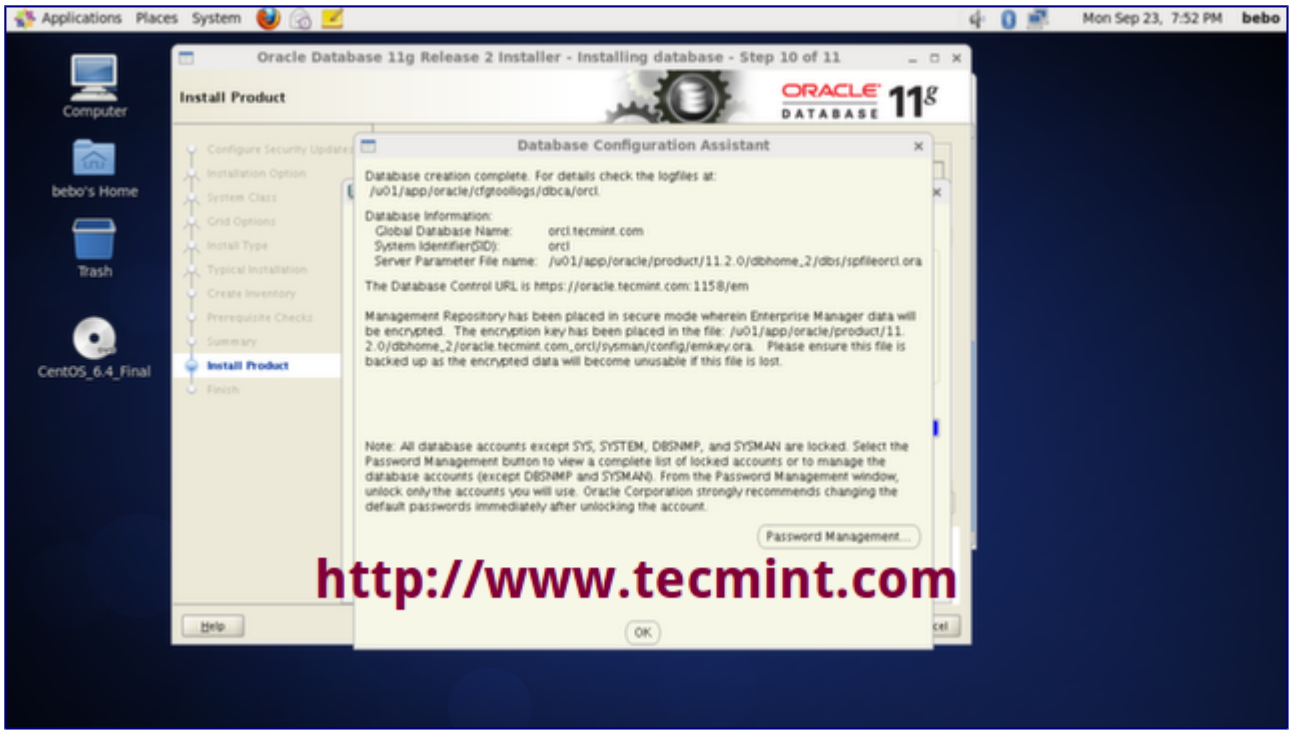
Product Install

15. Copying **database** files



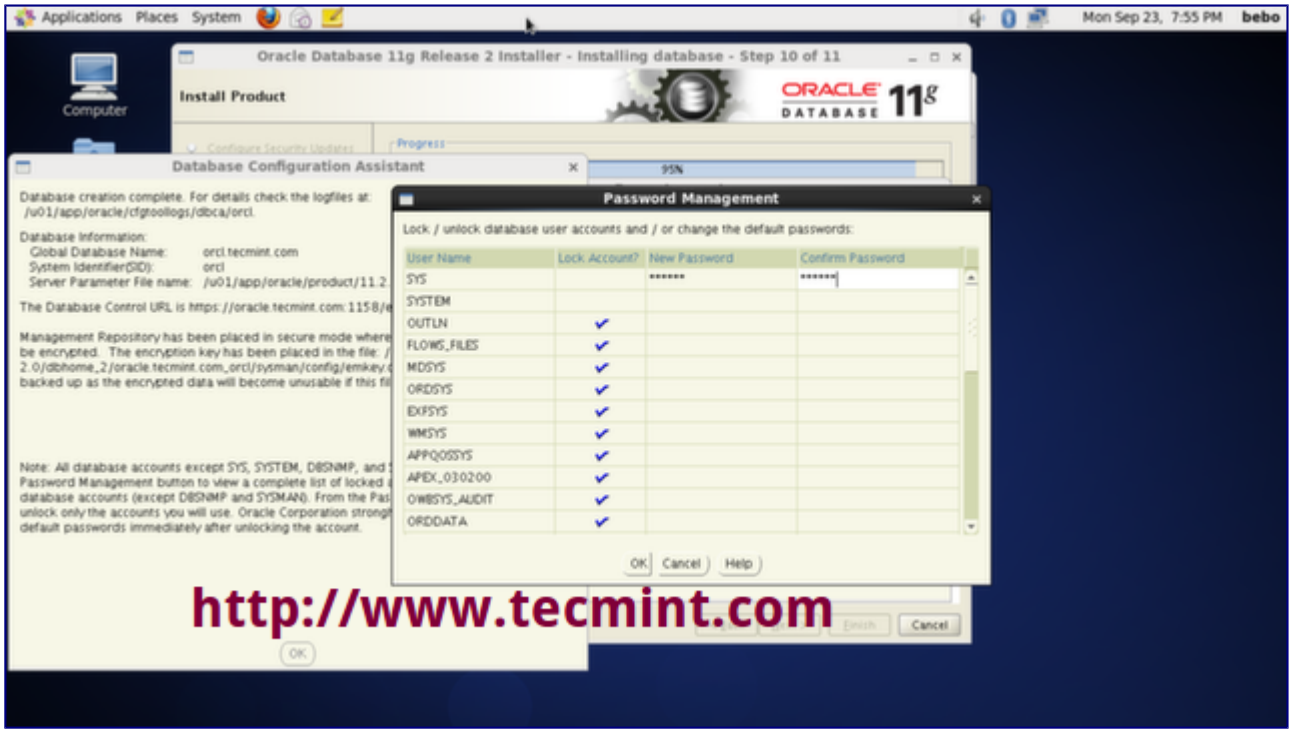
Copying Database Files

16. Click on "**Password Management**".



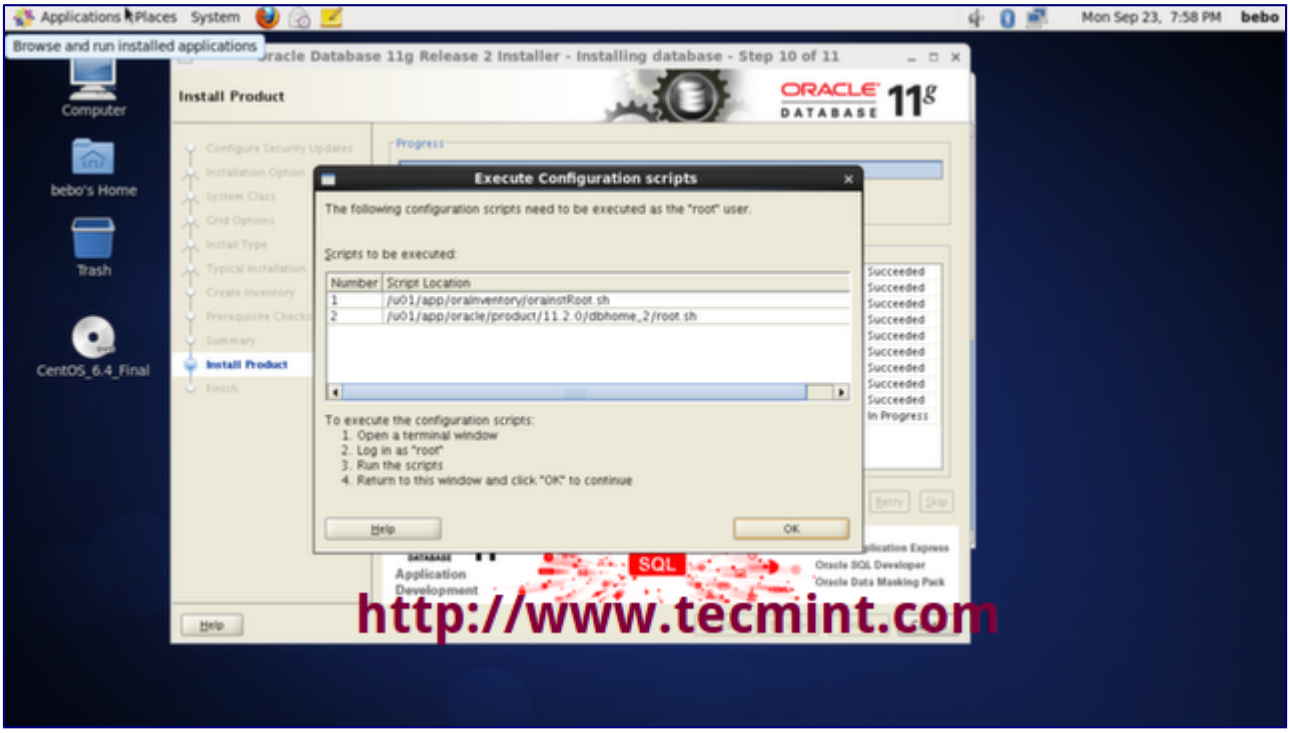
Oracle Password Management

17. Set password for user "**SYS**" and click on **OK** to continue.



Set SYS User Password

18. Configuration scripts need to be executed as the "**root**" user. Go to the path given in the screen and execute the scripts one by one. Click on **'OK**' once scripts is executed.



Execute Configuration Scripts

After this step, we need to run the script as root before close the window.

```
[root@oracle]# cd /u01/app/oraInventory
[root@oracle oraInventory]# ./orainstRoot.sh
Changing permissions of /u01/app/oraInventory.

Adding read,write permissions for group.

Removing read,write,execute permissions for world.



Changing groupname of /u01/app/oraInventory to oinstall.

The execution of the script is complete.
```

```
[root@oracle]# cd /u01/app/oracle/product/11.2.0/dbhome_2/
[root@oracle dbhome_2]# ./root.sh
```

```
Running Oracle 11g root.sh script...

The following environment variables are set as:
    ORACLE_OWNER= oracle
    ORACLE_HOME=  /u01/app/oracle/product/11.2.0/dbhome_2

Enter the full pathname of the local bin directory: [/usr/local/bin]:
   Copying dbhome to /usr/local/bin ...
   Copying oraenv to /usr/local/bin ...
   Copying coraenv to /usr/local/bin ...

Creating /etc/oratab file...
Entries will be added to the /etc/oratab file as needed by
Database Configuration Assistant when a database is created
Finished running generic part of root.sh script.
Now product-specific root actions will be performed.
Finished product-specific root actions.
```

Configuration to enable oracle auto start every reboot

Create a file oracle service in folder /etc/init.d

```
vi /etc/init.d/oracle
```

Copy the content below to file oracle upper

```
#!/bin/bash
#
# Run-level Startup script for the Oracle Instance and Listener
#
### BEGIN INIT INFO
# Provides:          Oracle
# Required-Start:    $remote_fs $syslog
# Required-Stop:     $remote_fs $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Startup/Shutdown Oracle listener and instance
### END INIT INFO

ORA_HOME="/u01/app/oracle/product/11.2.0/dbhome_1"
ORA_OWNR="oracle"

# if the executables do not exist -- display error

if [ ! -f $ORA_HOME/bin/dbstart -o ! -d $ORA_HOME ]
then
        echo "Oracle startup: cannot start"
        exit 1
fi

# depending on parameter -- startup, shutdown, restart
# of the instance and listener or usage display

case "$1" in
        start)
                # Oracle listener and instance startup
                echo -n "Starting Oracle: "
                su - $ORA_OWNR -c "$ORA_HOME/bin/dbstart $ORA_HOME"
                su - $ORA_OWNR -c "$ORA_HOME/bin/lsnrctl start"

                #Optional : for Enterprise Manager software only
                su - $ORA_OWNR -c "$ORA_HOME/bin/emctl start dbconsole"

                touch /var/lock/oracle
                echo "OK"
                ;;
        stop)
                # Oracle listener and instance shutdown
```

```
        echo -n "Shutdown Oracle: "

        #Optional : for Enterprise Manager software only
        su - $ORA_OWNR -c "$ORA_HOME/bin/emctl stop dbconsole"

        su - $ORA_OWNR -c "$ORA_HOME/bin/lsnrctl stop"
        su - $ORA_OWNR -c "$ORA_HOME/bin/dbshut $ORA_HOME"
        rm -f /var/lock/oracle
        echo "OK"
        ;;
reload|restart)
        $0 stop
        $0 start
        ;;
*)
        echo "Usage: $0 start|stop|restart|reload"
        exit 1
esac
exit 0
```

Change the mode for file oracle and set to auto start service oracle

```
su
chmod 755 /etc/init.d/oracle
chkconfig oracle on
```

2. Install Apache and PHP

Apache install

```
yum install httpd mod_ssl
```

Firewall

Notice that in some versions of CentOS, a firewall is installed by default which will block access to port 80, on which Apache runs. The following command will open this port:

```
iptables -I INPUT -p tcp --dport 80 -j ACCEPT
service iptables save
service iptables restart
```

Enable service apache auto start every restart

```
chkconfig httpd on
```

PHP5 Install

Let's move on to the PHP5 install. I'm not going to install all the modules available, just a few common ones so you get the idea.

As before, due to using yum to install PHP5, any dependencies are taken care of:

```
yum install php-mysql php-devel php-gd php-pecl-memcache php-pspell php-snmp php-xmlrpc php-xml
```

Once done, reload Apache:

```
service httpd restart
```

If you want to install all PHP libs to server, we can use command

```
yum install php-*
```

3. Install PDO & OCI8

3.1 Dependencies

Development packages

```
$ sudo yum install php-pear php-devel zlib zlib-devel bc libaio glibc

$ sudo yum groupinstall "Development Tools"
```

InstantClient

Download Oracle InstantClient RPM files here. Put these files in your server. Download the basic and devel packages.

Basic: oracle-instantclient11.2-basic-11.2.0.3.0-1.x86_64.rpm

Devel: oracle-instantclient11.2-devel-11.2.0.3.0-1.x86_64.rpm

Install the downloaded rpm files:

```
$ sudo rpm -ivh oracle-instantclient11.2-basic-11.2.0.3.0-1.x86_64.rpm
```

```
$ sudo rpm -ivh oracle-instantclient11.2-devel-11.2.0.3.0-1.x86_64.rpm


$ sudo ln -s /usr/include/oracle/11.2/client64 /usr/include/oracle/11.2/client

$ sudo ln -s /usr/lib/oracle/11.2/client64 /usr/lib/oracle/11.2/client
```

Create a file inside /etc/profile.d named oracle.sh and put this as the content:

```
export LD_LIBRARY_PATH=/usr/lib/oracle/11.2/client64/lib
```

And run it so we'll have LD_LIBRARY_PATH as an environment variable.

source /etc/profile.d/oracle.sh

If Instant Client is installed from ZIP files, make sure to create the library symbolic link first, for example

```
ln -s libclntsh.so.11.1 libclntsh.so
```

3.2 PDO_OCI

Download the PDO_OCI source using pecl.

```
$ pecl download PDO_OCI
$ tar -xvf PDO_OCI-1.0.tgz
$ cd PDO_OCI-1.0
```

Inside the PDO_OCI-1.0 folder, edit the file named config.m4.

Find a pattern like this near line 10 and add these 2 lines:

```
elif test -f $PDO_OCI_DIR/lib/libclntsh.$SHLIB_SUFFIX_NAME.11.2; then
  PDO_OCI_VERSION=11.2
```

Find a pattern like this near line 101 and add these lines:

```
11.2)
  PHP_ADD_LIBRARY(clntsh, 1, PDO_OCI_SHARED_LIBADD)
  ;;
```

Build and install the extension.

```
$ phpize
$ ./configure --with-pdo-oci=instantclient,/usr,11.2
$ make
$ sudo make install
```

To enable the extension, add a file named pdo_oci.ini under /etc/php.d and put this as the content:

```
extension=pdo_oci.so
```

Validate that it was successfully installed.

```
$ php -i | grep oci
```

You should see something like this in the output:

```
/etc/php.d/pdo_oci.ini,
PDO drivers => oci, odbc, sqlite
```

3.3 OCI8

Download the OCI8 source using pear

```
$ pear download pecl/oci8
$ tar -xvf oci8-1.4.9.tgz
$ cd oci8-1.4.9
```

Build and install the extension.

```
$ phpize
$ ./configure --with-oci8=shared,instantclient,/usr/lib/oracle/11.2/client64/lib
$ make
$ sudo make install
```

To enable the extension, add a file named oci8.ini in /etc/php.d with this content:

```
extension=oci8.so
```

Validate that it was successfully installed.

```
$ php -i | grep oci8
```

You should see something like this:

```
/etc/php.d/oci8.ini,
oci8
oci8.connection_class => no value => no value
oci8.default_prefetch => 100 => 100
oci8.events => Off => Off
oci8.max_persistent => -1 => -1
oci8.old_oci_close_semantics => Off => Off
oci8.persistent_timeout => -1 => -1
oci8.ping_interval => 60 => 60
```

```
oci8.privileged_connect => Off => Off
oci8.statement_cache_size => 20 => 20
```

3.4 Installing OCI8 from PECL

The OCI8 extension can be added to an existing PHP installation either automatically or manually from » PECL.

Run

```
pecl install oci8
```

For a manual install when the pecl command is not available, download the PECL OCI8 package, e.g. oci8-1.4.10.tgz.

Extract the package:

```
tar -zxf oci8-1.4.10.tgz
cd oci8-1.4.10
```

Prepare the package:

```
phpize
```

Configure the package, either using $ORACLE_HOME or Instant Client

```
./configure -with-oci8=shared,$ORACLE_HOME
```

or

```
./configure -with-oci8=shared,instantclient,/path/to/instant/client/lib
```

Install the package:

```
make install
```

After either an automatic or manual install, edit your php.ini file and add the line:

```
extension=oci8.so
```

Make sure the php.ini directive extension_dir is set to the directory that oci8.so was installed in.